Supplementary Problems for **Introduction to Parallel Computing**

Anshul Gupta

July 28, 1994

ASSIGNMENT for Chapters 1, 2

1. Given the choice of selecting the granularity of processors in a parallel computer (irrespective of the cost), would you chose fewer faster processors or a large number of slower processors. Justify your answer in terms of expected performance.

2. Consider the routing of messages in a parallel computer that uses store-and-forward routing. Assume that the given message of m words is now broken into k parts. These are now sent one by one across the store-and-forward network. A message can be sent as soon as the previous message has reached the next node in the path. Derive the time for transferring a message of size m to a processor l hops away. Comment on the time as the value of k varies between 1 and m.

3. The concept of a minimum congestion mapping is very useful. Consider two parallel computers with different interconnection networks such that a congestion-r mapping of the first into the second exists. Ignoring the dilation of the mapping, if each communication link in the second computer is more than r times faster than the first computer, the second computer is strictly superior to the first.

   Now consider mapping a d-dimensional hypercube onto a 2^d-processor mesh. Ignoring the dilation of the mapping, what is the minimum-congestion mapping of the hypercube onto the mesh? Use this result to determine whether a 1024-processor mesh with communication links operating at 25 million bytes per second is strictly better than a 1024-processor hypercube (whose processors are identical to those used in the mesh) with communication links operating at two million bytes per second.

4. Indicate whether the following statements are true or false. Supplement your answers with a brief explanation (a couple of lines).

   - Coarse grain parallel computers are less expensive than fine grain parallel computers offering identical performance because they need less communication hardware.
   - A star connected network is superior to a mesh network since the maximum distance between any two processors is only two.

- It is not possible to embed a mesh into a hypercube so that each link in the mesh is mapped to exactly one hypercube link.
- The message transfer time for store-and-forward and cut-through networks is identical for communication between neighboring processors.
- Data locality is critical only for message passing computers and not for shared address space computers because shared address space computers have hardware that makes the whole memory globally accessible.
- A message passing architecture can easily emulate a shared memory paradigm, but the reverse is not true.
- It is easier to connect 1024 processors in a hypercube network than a three-dimensional mesh.
- Dynamic interconnection networks can only be used to connect processors to memory banks. They cannot be used to connect processors to other processors.

Additional practice questions (not to be turned in): All problems in the sample problem set for Chapters 1 and 2, Problems 2.5, 2.7, 2.15 from the text.

ASSIGNMENT for Chapters 5, 6, 7

1. **10 points** Consider the problem of performing Gaussian Elimination on a mesh. Assume that the mesh has $\sqrt{n} \times \sqrt{n}$ processors. You need to find the upper triangular form of a matrix A. Consider 2 different mappings of the matrix on the mesh of processors.

   In one mapping each row of matrix is mapped to one processor (there are n rows in the matrix and n processors in the mesh). *In the second mapping* $\sqrt{n} \times \sqrt{n}$ subblocks of the matrix are mapped on to the mesh in a natural fashion.

   For each mapping do the following:

   Describe the parallel algorithm **(non pipelined)**. You can use a combination of text and picture to describe it. Derive the communication and computation time of this algorithm.

2. **5 points** Illustrate the bitonic sort on the following sequence
   5 3 7 0 2 1 4 6.

   Show the result of each step in the construction of the bitonic sequence and the merge operation that results in a sorted sequence.

3. **10 points** Consider the implementation of parallel quicksort of n numbers on a p processor hypercube. The pivot selection is based on the median of elements assigned to an arbitrarily selected processor in a subcube. An asymptotic analysis of this algorithm is presented in Section 6.4.1 of the text. Compute precise expressions for parallel runtime, speedup and efficiency of this algorithm using parameters t_c, t_s, t_h and t_w, as defined in the previous problem (assume a cut-through routed network). You may assume that the pivot selection is optimal (*i.e.,* each split results in two partitions containing equal numbers of elements). Does this time change for a store-and-forward network? Justify your answer.

4. **10 points** Assume a naive implementation of the quicksort algorithm on a hypercube. All n elements are initially assigned to processor P_0. A binary tree rooted at processor P_0 is embedded into the hypercube. This processor selects a pivot, retains all elements less than the pivot and assigns all elements greater than the pivot to the right son of the processor in the tree. This process continues until the elements are assigned to leaf processors. After this, the elements are sorted locally.

   Is this parallel formulation correct (will it yield a sorted sequence)? Compute the parallel runtime, speedup and efficiency for sorting n numbers on a p processor hypercube using this formulation. Is this formulation cost optimal? What is the relation between n and p if the efficiency is fixed at $E = 0.5$. You may assume that $t_s = t_h = 0$, and $t_w = t_c = 1$.

ASSIGNMENT for Chapters 7, 8, 9

1. **[10 points]** Compute the parallel run time, the speedup, and the efficiency of the parallel formulation of the connected-component algorithm presented in Section 7.6 for a p-processor mesh with store-and-forward routing and with cut-through routing. Comment on the difference in the performance of the two architectures. What is the difference between the performance on a mesh and a hypercube, both with CT routing?

   (Problem 7.13 of the text)

2. **[10 points]** In the analysis of various load-balancing schemes presented in Section 8.4.2, we assumed that the cost of transferring work is independent of the amount of work transferred. However, there are problems for which the work-transfer cost is a function of the amount of work transferred. Examples of such problems are found in tree-search applications for domains in which strong heuristics are available. For such applications, the size of the stack used to represent the search tree can vary significantly with the number of nodes in the search tree.

   Consider a case in which the size of the stack for representing a search space of w nodes varies as $\log w$. Analyze the performance of the GRR scheme on a hypercube.

   (Part of Problem 8.4 of the text)

3. **[10 points]** The theoretical analysis in the text shows that the isoefficiency function of RP is $O(p \log_{\frac{1}{1-\alpha}} p \times log_2^2 p)$. Here, α is a measure of the quality of the splitting function and varies between 0 and 0.5. The expression $\log_{\frac{1}{1-\alpha}} p$ is minimized when $\alpha = 0.5$. This can be verified intuitively. A value of $\alpha = 0.5$ represents a half split of the work at each step. Clearly, it is possible to balance load with the minimum number of work requests if each work split is a half split. As the value of α tends to 0, it implies a larger variability in the sizes of the work pieces. This would imply that more work requests are required to balance load. The value of α is a function of the application being solved. For some applications, it is possible to get good splitting functions (α closer to 0.5), whereas for others, α is low. It is therefore important to study the impact of the value of α on the scalability of various schemes. This is because different schemes may be impacted differently by changes in α.

   Estimate the effect of quality of splitting function (α) on the scalability of RP. Specifically, what happens to the scalability when α is reduced by a factor of 2.

   Use the following expansions to simplify your derivations:

$$\log_r p = \frac{\log_2 p}{\log_2 r}$$

$$\frac{1}{1-\alpha} = 1 + \alpha + \alpha^2 + \ldots$$

   and for $\alpha \ll 1$, $\frac{1}{1-\alpha} \sim 1 + \alpha$. Also,

$$\log_e(1+\alpha) = \alpha - \frac{\alpha^2}{2} + \frac{\alpha^3}{3} \ldots$$

Again, for $\alpha << 1$, we can neglect higher powers of α, and

$$\log_e(1+\alpha) \sim \alpha$$

.

4. **[10 points]** Consider the parallel algorithm for solving the 0/1 knapsack problem on a hypercube in Section 9.1.2. Map this algorithm onto a mesh of $\sqrt{p} \times \sqrt{p}$ processors. During iteration i, a processor computes c/p elements of row i of table F. Derive the parallel run time, speedup, and efficiency for this algorithm. Show that the algorithm is not cost-optimal.

   The algorithm is not cost optimal because there is no locality in data reference. However, if we can ensure that all the weights are below a certain threshold k ($k < \sqrt{p}$), then it is possible to ensure a measure of data locality. Assuming that weights of all objects are below a certain value k, compute the parallel run time, speedup, and efficiency in this case. Show that, under this assumption, the algorithm is cost-optimal.

5. **[10 points]** The parallel formulation of the longest-common-subsequence problem presented in the text has an upper bound of 0.5 on its efficiency. What is the runtime of the algorithm for solving the longest-subsequence-problem on a p processor hypercube (with CT routing) using cyclic-striped mapping (in terms of t_s, t_w and t_c)? What is the runtime if the partitioning technique used is the block-cyclic mapping discussed in Section 5.1.1. Do these formulations have an upper-bound on the efficiency?

Additional practice questions (not to be turned in): Problem 7.14, 8.2, 8.4, 8.5, 8.8, 9.1, 9.9.

CSci 5151 - Winter-1992
Homework # 1

1. Multiple choice problems.

(a) Which one of the following statements is FALSE ?

i) PEs are synchronized at instruction level in an MIMD computer.

ii) In an MIMD machine, each PE must have a Control Unit.

iii) The network providing access to shared memory is usually complex and a large fast network can be very expensive.

iv) In a non shared memory machine, PEs communicate through an interconnection network.

(b) In which of the following networks, the Maximum Bandwidth per processor decreases as the number of processors in the configuration is increased ?

i) Crossbar.

ii) Bus.

iii) Hypercube.

iv) Multistage.

(c) In which of the following networks does the total wire length (used in the interconnection network) per processor remain constant ?

i) Hypercube.

ii) Multistage.

iii) A wrap around mesh.

iv) A 3-D mesh.

2. Which of the following statements are TRUE ? Give a one line justification for your answer, whether the statement is true or false. Credit will be given only if the justification is correct.

(a) If enough effort and money is spent, it is possible to construct an arbitrarily fast sequential processor.

(b) A parallel machine with 100 processors of 100 MIPS each will deliver ten times the throughput of a machine with a single 1000 MIPS processor on an application.

(c) It is usually easier to convert a sequential algorithm to a parallel algorithm for a shared memory architecture than for a message passing architecture.

(d) Commercially available shared memory computers often allow each processor some local memory also which is directly accessible only from its parent processor.

(e) The main challenge of parallel processing is to build massively parallel computers rather than to exploit their computational power.

(f) It is costlier to build an MIMD compueter than an SIMD computer with the same number of processors (of comparable speed).

(g) A message passing architecture can easily emulate a shared memory paradigm, but the reverse is not true.

(h) Using caches can help in reducing the load on the network providing access to shared memory.

(i) Shared memory architectures readily support rapid access by all the processors to large dynamically changing data structures.

3. Map nodes of a complete binary tree with four levels ($2^4 - 1$ nodes) to nodes of a four-dimensional hypercube so that as many tree edges as possible map onto hypercube edges. Labelled diagrams of a tree d a hypercube is acceptable.

   Hint: In the optimal mapping, only one tree edge corresponds to two hypercube edges.

4. Given $n = 2^{k/2}$ for a positive even integer k, write a function that maps elements of an nxn mesh with wraparound connections into nodes of a k-dimensional hypercube so that all mesh connections map directly onto connections in the hypercube. The function should take as input two integers i and j ($0 \leq i \leq n$, $0 \leq j \leq n$) which are the coordinates of the location of the processor in the mesh, and should yield as output the number of the corresponding processor in the hypercube. For example in the mapping (see Fig. 1) of a 16-processor mesh to a hypercube, given the values 1,2 the function should yield the value 7.

Figure 1: Mapping of a Mesh on a Hypercube (problem 4).

CSci 5299 Winter '91

HW # 1

1. It is required to determine the largest of a set of n numbers. Give optimal algorithms to solve this problem on multicomputers using the following interconnection networks :

a) A linear array of processors.

b) A mesh.

c) A hypercube.

d) A tree.

Assume that a comparison between two number takes time t_c and sending one number from a processor to one of its nearest neighbors takes time t_b. For each of the above, derive expressions for Parallel Run Time, Efficiency and Iso-efficiency in terms of n, p, t_c and t_b, where p is the number of processors used.

2. Problem # 1.11 from Page 31 in Akl's book.

3. In an SIMD computer, one of the p processors holds a datum that it wishes to make known to every other processor in the interconnection network. Show how this can be done for each of the interconnections in Problem # 1 above in minimum possible time. Derive an expressions for the Parallel Run Time in each case. Assume that sending the number from a processor to one of its nearest neighbors takes time t_b.

4. Consider the following simple implementation of matrix multiplication algorithm. Consider a p processor wrap-around mesh (with sqrt(p) rows and sqrt(p) columns) on which two n X n matrices are to be multiplied, where n >= sqrt(p). Each processor stores O(n^2/p) data elements. As the first step of the algorithm, each processor acquires all those elements of both the matrices that are required to generate the n^2/p elements of the product matrix which are to reside in that processor. In order to do so, each processors has to get all the elements of one matrix stored in all the processors in its row in the mesh and all the elements of the other matrix stored in all the processors in its column in the mesh.

In the second step of the algorithm, each processor computes n^2/p elements of the product matrix from elements of the input matrices which it has acquired after the communication step.

For example if two 8 X 8 matrices A and B are to multiplied using 16 processors, P_0 to P_15, then initially, each processor will hold 4 elements of each matrix. For instance, P_0 will hold A(0,0), A(0,1), A(1,0), A(1,1), B(0,0), B(0,1), B(1,0) and B(1,1). The configuration of the processors is shown in the Figure. After the communication step, P_0 will contain A(0,0) to A(0,7), A(1,0) to A(1,7), B(0,0) to B(7,0) and B(0,1) to B(1,7). In order to get all this data, P_0 will have to communicate with P_1, P_2, P_3, (i.e. the processors in the same row) and P_4, P_8, P_12 (i.e. the processors in the same column).

Assume that a multiplication and addition operation takes time t_c and sending one number from a processor to one of its nearest neighbours takes time t_b. Apart from this each message sent from one processor to another incurs an additional cost equal t_s which we call the message startup time.

a) Describe an efficient way of affecting the data communication and compute the total communication time that you scheme requires.

b) Compute the Parallel Run Time, Speedup and Iso-efficiency for

the algorithm in terms of p, n, t_c, t_b and t_s.

c) Compute the Parallel Run Time, Speedup and Efficiency for

i) n = 1024, p = 64, t_c = 1, t_b = 1, and t_s = 10.
ii) n = 1024, p = 64, t_c = 1, t_b = 10, and t_s = 10.
iii) n = 64, p = 64, t_c = 1, t_b = 1, and t_s = 10.
iv) n = 64, p = 64, t_c = 1, t_b = 10, and t_s = 10.

d) Compute the exact Iso-efficiency function for the values of

t_c, t_b and t_s in parts (i) and (ii) above if the efficiency has

to be maintained at

i) 25%.
ii) 75%.

CSci - 5299   HW # 2
^^^^^^^^^^^   ^^^^^^

Due on April 26, 1990.
~~~~~~~~~~~~~~~~~~~~~

1.(a)

Develop an Algorithm to compute Prefix Sums of "n" integers on a mesh of "p" processors (sqrt(p) X sqrt(p) mesh). Compute its parallel run time, speedup, efficiency and iso-efficiency.

While doing the above analysis, assume that each computation step takes time "t_e" and the communication of one integer from a processor to another one which is "h" hops away takes "t_c X h" time.

(Hint : Refer to section 13.2.4 of Akl's book)

(b)

Compute speedup and efficiency for the following set of values of parameters :-

(i)   t_e = 1; t_c = 1 ; n = 1024 ; p = 64.
(ii)   t_e = 1; t_c = 10; n = 1024 ; p = 64.
(iii) t_e = 1; t_c = 10; n = 16384; p = 64.

2. Refer to the FFT algorithm in the paper "On the Scalability of FFT on Parallel Computers". Assume the sequential run time of the algorithm to be "t_e X n X log(n)", i.e. each computation step takes time "t_e". Let the time taken to transfer a message of size "m" between two processors that are "h" hops apart be "t_c X m X h". Now compute the following for an "n" point FFT on "p" processors (assume both "n" and "p" are powers of 2 and "p" <= "n") : -
~~~~~~~~~~~~~~~~~~~~~

(i) Parallel run time, Speedup, Efficiency and Iso-efficiency on a linear array of processors.

(ii) Parallel runtime, Speedup, Efficiency and Iso-efficiency on a mesh connected multicomputer.

(iii) Parallel runtime and Speedup for n = 1024; p = 64; t_e = 1; and t_c = 0.5 on a hypercube.

(iv) The Iso-efficiency function for a hypercube for maintaining an efficiency of 75% with the set of values for "t_e" and "t_c" as given in part (iii) above.

Sci 5299 Winter '91

Homework # 2. Due date Feb 28, 1991.

1\. Refer to the Odd-Even sorting algorithm in Section 4.3 of Akl's book. Consider the case when there are more than one elements per processor; i.e., assume that "n" numbers are to be sorted using "p" processors connected in a linear array where "n" > "p". In the first step, each processor locally sorts all the "n/p" elements that it stores. In each of the next "p/2" steps the processor pairs communicate (in the same manner as the one element per processor case), exchange all their data, merge the two lists and one processor stores the top half while the other one stores the lower half. Assume that the initial local sorting phase takes time "t_c n/p log(n/p)", each merging of 2 lists of "n/p" numbers takes "2t_c n/p" time and each data exchange along the bidirectional link between a pair of processors takes "t_s + t_b n/p" time.

(a) What should be the restriction on the number of processors "p" so that the algorithm is efficient or cost optimal ? Assume t_s = 0 for this part.

(b) Derive expressions for Parallel Execution Time, Speedup, and Efficiency in terms of "n", "p", "t_c", "t_s" and "t_b".

(c) Derive an expression for the Iso-efficiency function in terms of "n", "p", "t_c" and "t_b". Assume t_s = 0 for this part.

(d) Write a small computer program to generate tables for values of efficiency (Use the format of Table 18.1 on Page 339 of fox's book) using the expression derived for it in part (b) above. Assume that t_c = 1. Choose the values of "t_b" from {2, 5, 10} "t_s" from {5, 10, 50}. Thus there should be 3 X 3 = 9 tables. For each table, "n" takes values from 1K (i.e. 1024) to 2048K and "p" takes values of 2, 4, 8, 16, 32, 64 and 128 as in Table 18.1 of the book

.

2. Solve parts (a), (b), (c) and (d) above for the variation of the algorithm for implementation on a hypercube as given in Section 18.4 of Fox's book. (Copies of chapter 18 were distributed in the class.) The cost of performing each step remains the same but "1 + 2 + ... + log(p)" steps are now performed instead of "p/2" steps as in 1. above.

3. Problems 4.31 and 4.32 on Pages 106, 107 of Akl's book.

Hints for part (d) of Prob # 1 and 2 above :

i. Construct a 9 X 7 X 12 array of floating point numbers.

ii.
```
For i = 1 to 9 do
begin
        read in a pair of values for "t_b" and "t_s" from a file
          that contains 9 such pairs;
        For j = 1 to 12 do
        begin
                n = 1024 X 2^(j - 1);
                For k = 1 to 7 do
                begin
                        p = 2^k;
                        if p <= n then
                        begin
                                plug the values of n, p, t_b, t_s and
                                    t_c (use 1) in the expression for E;
                                Array[i,k,j] = E;
                        end
                        else Array[i,k,j] = 0.0;
                end;
        end;
        Print Array[i, 1..7, 1..12];
end.
```

Sci 5299 Winter 1991

HW # 3 Due Date :

1. Briefly describe an implementation of an algorithm for finding All Sums (refer to page 46 of Akl's book) on a single stage shuffle exchange network. Derive expressions for Parallel Execution Time, Speedup, Efficiency and Iso-efficiency Function in terms of "n" and "p", where "n" is the total number of data items and "p" is the number of processors to be used.

2. Briefly describe an implementaion of the striped version of Floyd's algorithm for finding the shortest paths (refer to Sahni's paper distributed in the class) on a mesh. Assume that the graph has "n" nodes and a SQRT(p) X SQRT(p) mesh is being used. Derive expressions for Parallel Run Time, Speedup, Efficiency and Iso-efficiency Function in terms of "n" and "p".

3. Briefly describe an algorithm to multiply a "n" X "n" matrix with a "n" X "1" vector on a SQRT(p) X SQRT(p) processor mesh. Assume that

the input vector is distributed horizontally in the first row of the mesh processors and the resultant vector is also expected to lie in the first row of processors. Derive expressions for Parallel Run Time, Speedup, Efficiency and Iso-efficiency Function in terms of "n" and "p".

NOTE: Please use only order terms for all expressions in all the derivations in all the problems.

CSci 5151 - Winter -1992
Homework #3

Due on Thursday, Feb. 27 for class students
Due on Tuesday, March 3 for UNITE students
Due 1 week after receipt of this homework by NTU students

1. An n X n matrix is stored on a wrap-around n X n mesh such that each processor contains exactly one element initially. Each processor is required to get all the elements that reside on the same row and column as the element it initially contains. For example, processor (i,j) should get all elements of row i and column j. Describe a parallel algorithm to do this in O(n) time. Can this problem be done on a hypercube in less time ? In this case assume that n = 2^q, and the n X n matrix is stored on a 2^(2q) processor hypercube.

2. Consider the problem of multiplying an n X n matrix with an n X 1 vector on a mesh connected multicomputer.

   1) Briefly describe a parallel algorithm for p = n^2. Here each processor initially has one element of the matrix and each processor of first row has one element of the vector. What is the parallel execution time in terms of n, p, t_c (time to perform one addition and one multiplication), t_s and t_w (sending a message of size m between two neighboring processors takes (t_s + m X t_w) time) ? Is the algorithm cost optimal ?

   2) Describe the parallel algorithm for the case p < n^2. Clearly describe how the elements of the matrix and the vector are mapped on to the processors. Calculate the parallel time, speedup and efficiency in terms of n, p, t_c, t_s and t_w. What is the iso-efficiency function ? For what values of n (in terms of p) is this algorithm cost optimal ?

3. 1) Derive the exact expressions for parallel execution time, speedup, efficiency and isoefficiency function for Akl's matrix multiplication algorithm described in section 4.2 of the report for an MIMD wrap-around mesh connected multicomputer, in terms of n, p, t_c, t_w and t_s, where t_c, t_s and t_w are defined above. You can simplify (sqrt(p) - 1) to sqrt(p).

   2) Will the parallel execution time of Akl's algorithm change if the algorithm is implemented on a hypercube instead of a wrap-around mesh?

   3) How does the run time of Akl's algorithm compare with that of the simple matrix multiplication algorithm described in section 4.1 for wraparournd mesh ? Which method are you more likely to implement for large matrices ? Why ?

4. State whether the following statements are true or false. Please provide brief explanations.

   i) It is possible that a cost optimal parallel algorithm may not have an iso-efficiency function. An algorithm is defined to be cost optimal if the p X T_p product is of the same order as the sequential execution time.

   ii) It is possible that a parallel algorithm having an iso-efficiency function may not be cost optimal for some value of problem size and the number of processors.

   iii) All parallel algorithms will have asymptotically better (smaller) iso-efficiency functions on a hypercube than on a wrap-around mesh.

   iv) If a parallel algorithm is cost optimal on a tree connected architecture where processors are only at leaves, then it has to be cost optimal on a hypercube.

   v) If a parallel algorithm is cost optimal on a hypercube, then it has to be cost optimal on a tree connected architecture where processors are only at leaves.

CSCI 5151, HW # 4 (Winter 1992)

Due on March 12, 1992

1. Refer to the Odd-Even sorting algorithm in Section 4.3 of Akl's book. Consider the case when there are more than one elements per processor; i.e., assume that "n" numbers are to be sorted on "p" processors connected in a linear array where "p" < "n". Assume "n" to be a multiple of "p". In the first step, each processor locally sorts all the "n"/"p" elements it has. In each of the next "p"/2 steps, processor pairs communicate (in the same manner as the one element per processor case), exchange all their data, merge the two lists. One processor stores the top half of this list and the other one stores the bottom half. Assume that the initial local sorting phase takes time equal to "tc" * "n"/"p" * log("n"/"p"), each merging of two lists of size "n"/"p" takes time equal to 2 * "tc" * "n"/"p", and each data exchange along the bidirectional link between a pair of processors takes time equal to "ts" + "tw" * "n"/"p".

(a) Assuming "ts" = 0, what is the range of "p", in terms of "n", for which the algorithm is cost optimal ?

(b) Derive expressions for Parallel Execution Time ("Tp"), Speedup ("S"), Efficiency ("E") and Iso-efficiency in terms of "n", "p", "tc", "tw" and "ts". Assume "ts" = 0 for the Iso-efficiency calculation.

(c) Use the small C program on Page 2 to generate tables for the values of efficiency using the expression derived for it in part (b) above. Assume "tc" = 1. Choose the values of "tw" from {2, 5} and the values of "ts" from {150, 250}. Thus there should be 2 X 2 = 4 tables. Each table will have 10 columns for "n" varying from 1 K to 512 K and 7 rows for "p" varying from 2 to 128.

2. Solve parts (a), (b) and (c) above for the variation of this algorithm for implementation on a hypercube as given in Section 18.4 of Fox's book. (Copies of chapter 18 are available at Campus Copies). The cost of performing each step remains the same but the number of steps to be

performed now is "1 + 2 + ... + log("p")" instead of "p"/2.

3. Consider Floyd's all pair shortest path algorithm on a mesh connected MIMD machine.

(a) Design a parallel version of the algorithm for a SQRT(n) X SQRT(n) mesh with store and forward routing, where "n" is the number of vertices in the graph. The adjacency matrix is partioned in checkerboard fashion and initially each processor contains SQRT(n) X SQRT(n) elements of the adjacency matrix. What is the parallel run time, speedup, and efficiency ?

(b) Do Part (a) for the striped version of the algorithm now and assume that each processor intially contains one column of the adjacency matrix. Note that the amount of data stored on each processor remains the same, only the mapping of the elements of the adjacency matrix on the processors is different. How does this variation of the algorithm compare with the previous version in terms of speedup and efficiency ?

***** THE FOLLOWING PARTS ARE FOR PRACTICE ONLY AND WILL NOT BE GRADED ****

(c) Do Part (a) on a SQRT(n) X SQRT(n) mesh with cut-through routing.

(d) Do Part (a) on a SQRT(p) X SQRT(p) mesh with store and forward routing, where "p" <= "n". What is the iso-efficiency in this case ?

HOMEWORK #1

CSci 5152, Spring 1994

1. Solve problems 4.6, 4.12, 7.1, and 7.9 from the textbook.

2. Derive expressions for the parallel run time of the Minimum Spanning Tree algorithm (Prim's Algorithm) on mesh and hypercube architectures. Assuming that the cost of a parallel architecture is proportional to the total number of inter-processor links, which of these architectures is more cost-effective for a parallel implementation of Prim's Algorithm?

3. Is it possible that a parallel algorithm for which the isoefficiency function exists may not be cost-optimal for certain values of problem size and number of processors? If not, give a proof. If yes, give an example of such an algorithm giving the range of problem size and the number of processors for which it is not cost-optimal.

Practice Problems: Not to be turned in

Solve problems 4.18, 5.24, 5.25, 5.26, 5.27, 7.10, and 7.13 from the textbook.

ASSIGNMENT # 2
CSci 5151
(Winter 1994)
Due Date: Feb 3 1994 (@2:30 PM)
Late submission penalty: 10% per day

1. Although a ring topology is hardly used in constructing practical parallel computers, why is it still important to study ring algorithms for performing various communication operations?

2. Give an example of a communication operation for which the best algorithm on a ring with CT routing is the hypercube algorithm. Also give an example of a communication operation for which the hypercube algorithm implemented on a ring with CT routing performs worse that the ring algorithm.

3. What is meant by the *dual* of a communication operation? What are the duals of the following operations?

   a. A simple one-to-one message transfer
   b. Single-node accumulation
   c. One-to-all broadcast
   d. All-to-all broadcast
   e. A single circular clockwise shift

4. Which of the following communication operations are faster on a hypercube with CT routing than on a hypercube with SF routing?

   a. A simple one-to-one message transfer
   b. One-to-all broadcast
   c. All-to-all broadcast
   d. Single-node accumulation
   e. Multi-node accumulation

5. Assume that $t_h = 0$. Which of the operations listed in Problem 4 above are no faster on a p-processor hypercube than on a $\sqrt{p} \times \sqrt{p}$ mesh with CT routing?

6. In which of the operations listed in Problem 4 the t_w term is the same on a p-processor hypercube as that on a $\sqrt{p} \times \sqrt{p}$ mesh with CT routing? What is the implication of this for very large messages?

7. Consider implementing the hypercube algorithm for one-to-all broadcast on a ring with SF routing. What is the communication time of this operation? What is the relation between t_s and t_w (in terms of m and p) for which one-to-all broadcast on a ring with SF routing is faster by using the hypercube algorithm rather than the standard ring algorithm depicted in Figure 3.2 of the text?

8. Describe an algorithm to perform single-node accumulation on a ring with CT routing. Draw a figure along the lines of Figure 3.6. What is the total communication time?

9. Solve problem 4.1 on page 141 of the textbook.

HW # 3
CSci 5152 (Spring 1994)
Due Date: June 2 1994 (@2:30 pm)
Late submission penalty: 20% per day

1. The serial run time of an n-point FFT computation is $t_c n \log n$. The parallel run time of the binary-exchange algorithm for the same problem is $t_c \frac{n \log n}{p} + t_s \log p + t_w \frac{n \log p}{p}$ on a p-processor hypercube.

   (a) If $t_c = 1, t_s = 10, t_w = 0.33$, write exact expressions for T_P, S, and E. What is the asymptotic isoefficiency function for $E = 0.75$?

   (b) Repeat part (a) for the cases in which $t_w = 0.5$ and $t_w = 1$, and everything else remains unchanged.

   (c) Comment on the impact of the ratio of t_w to t_c on the overall scalability of the algorithm on a hypercube.

2. Given a choice between the binary-exchange and the 2-D transpose algorithms, which one would you use for $n = 2^{12}$ and $p = 2^6$ in each of the following cases: $\frac{t_s}{t_w} = 250$, $\frac{t_s}{t_w} = 50$, $\frac{t_s}{t_w} = 10$, $\frac{t_s}{t_w} = 2$, $\frac{t_s}{t_w} = 0$.

3. Solve Problems 11.2 and 11.3 from the text.

   Note: Figure 11.29 and the hint for Problem 11.2 are incorrect as given in the text. Please refer to the errata to get the correct versions.

4. Solve Problem 11.9 from the text.

Practice Problems: not to be graded

1. Figure 11.7 in the text shows the block-tridiagonal matrix resulting from a 5-point finite difference mesh of size 4×4 with natural ordering. Draw the structure of the block-tridiagonal matrix resulting from a similar 3×5 finite difference mesh. Now draw the same for a 5×3 finite difference mesh.

2. Solve Problems 10.3, 10.4, 10.5, 11.8, 11.10, and 11.12 from the text.

ASSIGNMENT # 3
CSci 5151
(Winter 1994)
Due Date: Feb 17 1994 (@2:30 PM)
Late submission penalty: 10% per day

1. Solve problem 4.1 on page 141 of the textbook.

2. Which of the speedup curves shown in Figure 2 are realizable in practice? Give a brief explanation for your answer.

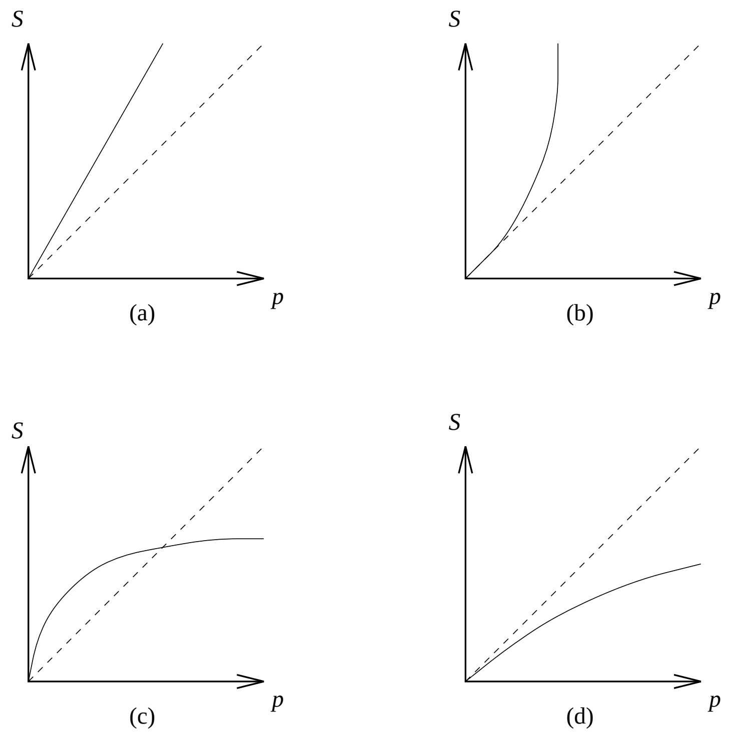

Figure 2: Choices for Problem 2.

3. Which of the following statements are TRUE ? Give a one line justification for your answer, whether the statement is true or false. Credit will be given only if the justification is correct.

    a. If enough effort and money is spent, it is possible to construct an arbitrarily fast sequential processor.

    b. It is costlier to build a parallel computer with 500 processors of 1 MFLOPS each than to construct a serial computer with a single 500 MFLOPS processor.

    c. The main challenge of parallel processing is to build massively parallel computers rather than to exploit their computational power.

d. Any parallel algorithm will have a asymptotically better (smaller) isoefficiency function on a hypercube than on a mesh.

e. Any problem of size W will run more efficiently on a p-processor hypercube than on a p-processor 2-D mesh (both with SF routing).

4. Solve Problems 4.8 and 4.9 from the textbook.

5. A drawback of increasing the number of processors without increasing the total work load is that the speedup does not increase linearly with the number of processors. Based on your experience with Problem 4.8 of the text, discuss whether using scaled speedup instead of standard speedup solves the problem in general.

6. Consider the multiplication of an $n \times n$ matrix with an $n \times 1$ vector on a $\sqrt{p} \times \sqrt{p}$ mesh with CT routing.

   a. What is the parallel run time for $p = n^2$ if the checkerboard mapping as shown in Figure 5.9 of the text is used?

   b. What is the parallel runtime for checkerboard mapping with $p = n$? Is the run time asymptotically different from the case when $p = n^2$?

   c. What is the parallel run time for $p = n$ if the striped mapping as shown in Figure 5.8 of the text is used?

7. Solve Problem 5.11 from the textbook.

Practice Problems (not to be turned in)

1. Which of the following statements are TRUE ?

   a. It is possible that an algorithm for which the isoefficiency function exists may not be cost-optimal for certain values of problem size and number of processors.

   b. Among two algorithms with asymptotically different isoefficiency functions, the one with the asymptotically smaller isoefficiency function will always yield a higher efficiency than the other one.

2. Solve Problems 4.10, 5.24–5.27, and 5.30–5.36 from the textbook.

Chapter 3, Supplementary Problems

1. Although a ring topology is hardly used in constructing practical parallel computers, why is it still important to study ring algorithms for performing various communication operations?

2. What is meant by the *dual* of a communication operation? What are the duals of the following operations?

   - A simple one-to-one message transfer
   - Single-node accumulation
   - One-to-all broadcast
   - k-to-all broadcast
   - All-to-all broadcast
   - A single circular clockwise shift

3. Give an example of a communication operation for which the best algorithm on a ring with CT routing is the hypercube algorithm. Also give an example of a communication operation for which the hypercube algorithm implemented on a ring with CT routing performs worse that the ring algorithm.

4. Which of the following communication operations are faster on a hypercube with CT routing than on a hypercube with SF routing?

   - A simple one-to-one message transfer
   - One-to-all broadcast
   - k-to-all broadcast
   - All-to-all broadcast
   - Single-node accumulation
   - Multi-node accumulation
   - Single-node gather
   - One-to-all personalized communication
   - All-to-all personalized communication
   - Circular shift

5. Assume that $t_h = 0$. Which of the operations listed in Problem 4 are no faster on a p-processor hypercube than on a $\sqrt{p} \times \sqrt{p}$ mesh with CT routing?

6. In which of the operations listed in Problem 4 the t_w term is the same on a p-processor hypercube as that on a $\sqrt{p} \times \sqrt{p}$ mesh with CT routing? What is the implication of this for very large messages?

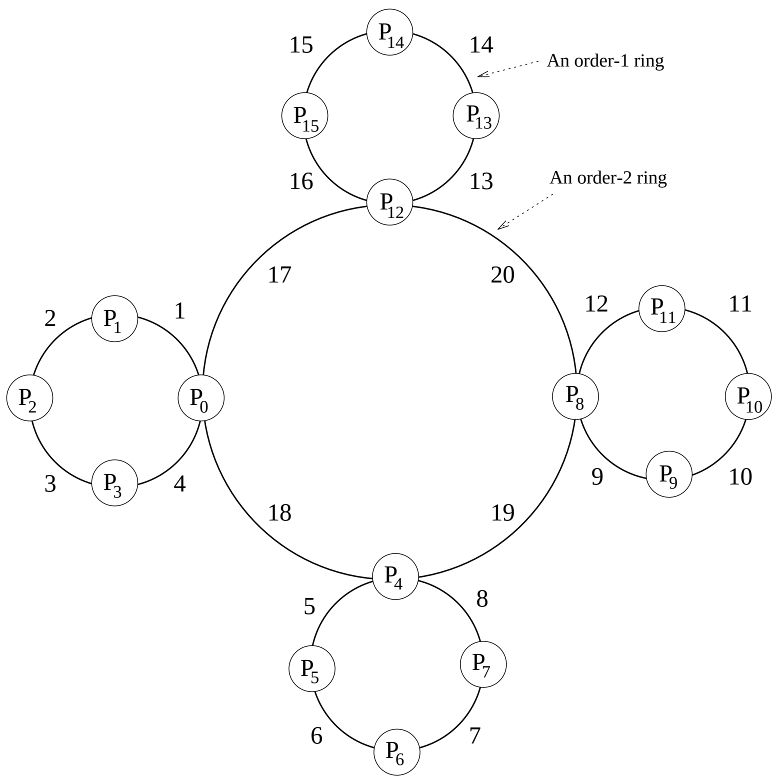

Figure 3: A 4-ary 2-ring.

7. Define a k-ary 1-ring is a ring network connecting k processors. A k-ary d-ring consists of k k-ary $(d-1)$-rings connected in a ring topology. This topology is also referred to as a **ring of rings**. A k-ary 1-ring is called an order-1 ring. A ring connecting rings of order-1 is called an order-2 ring. An order-i ring choses one of the k processors from each of the k (order-$(i-1)$) rings that it connects. For example, Figure 1 shows a 4-ary 2-ring, which has 16 processors and 20 communication links. Derive expressions for the total number of processors and links in a k-ary d-ring.

8. What is the maximum distance between any two processors (that is, the diameter) of a k-ary d-ring?

9. Describe an algorithm to perform one-to-all broadcast on a k-ary d-ring with SF routing. What is the total communication time of this operation?

10. Describe an algorithm to perform one-to-all broadcast on a k-ary d-ring with CT routing. What is the total communication time of this operation?

11. Describe an algorithm to perform single-node accumulation on a ring with CT routing. Draw a figure along the lines of Figure 3.6. What is the total communication time?

12. Show that any one-port algorithm to perform all-to-all broadcast, multinode accumulation, one-to-all personalized communication, or single-node gather will take at least $t_w m(p-1)$ time on any architecture.

13. Although, using all the communication ports of a processor simultaneously may improve the communication time of some operations, it is not likely to have a significant impact on the efficiency of parallel algorithms using these communication operations. Why?

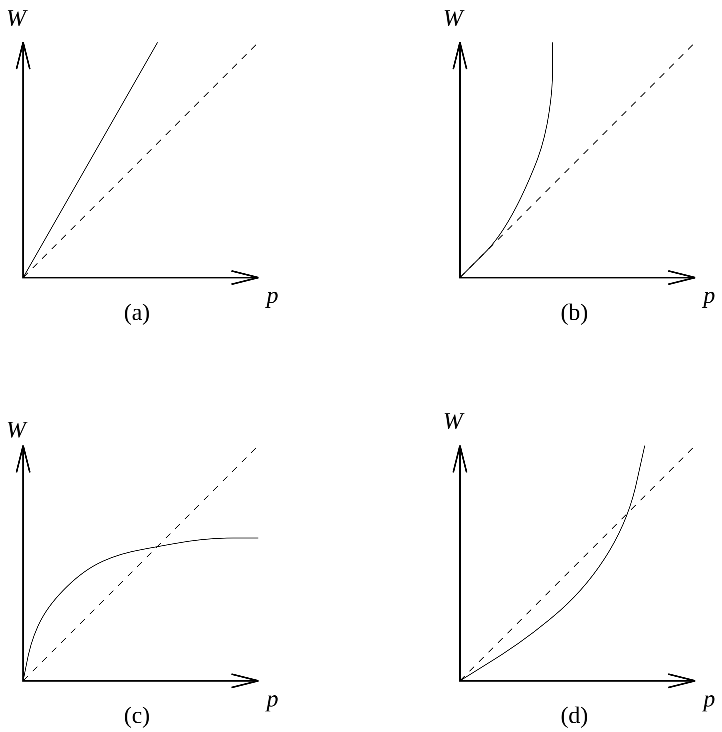

Figure 4: Choices for Problem 1.

Chapter 4, Supplementary Problems

1. Which of the isoefficiency curves shown in Figure 4 are realizable in practice? Give a brief explanation for your answer.

2. Which of the speedup curves shown in Figure 5 are realizable in practice? Give a brief explanation for your answer.

3. Which of the following statements are TRUE ? Give a one line justification for your answer, whether the statement is true or false. Credit will be given only if the justification is correct.

   - If enough effort and money is spent, it is possible to construct an arbitrarily fast sequential processor.
   - A parallel computer with 100 processors of 100 MIPS each will deliver ten times the throughput of a machine with a single 1000 MIPS processor on an application.
   - It is usually easier to convert a sequential algorithm to a parallel algorithm for a shared memory architecture than for a message passing architecture.
   - Commercially available shared memory computers often allow each processor some local memory also which is directly accessible only from its parent processor.
   - The main challenge of parallel processing is to build massively parallel computers rather than to exploit their computational power.

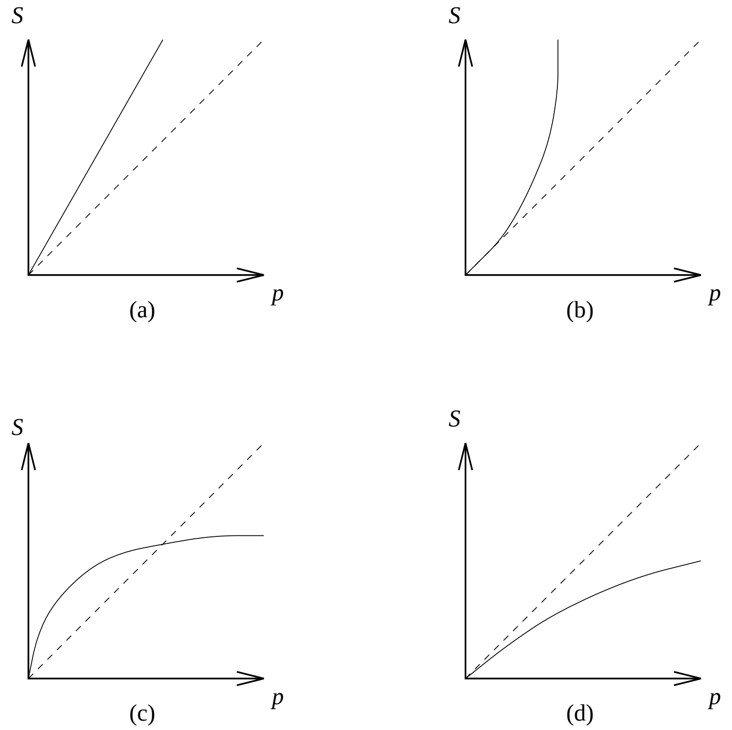

Figure 5: Choices for Problem 2.

- It is possible that an algorithm for which the isoefficiency function exists may not be cost-optimal for certain values of problem size and number of processors.
- Among two algorithms with asymptotically different isoefficiency functio ns, the one with the asymptotically smaller isoefficiency function will always yield a higher efficiency than the other one.
- Any parallel algorithm will have a asymptotically better (smaller) isoefficiency function on a hypercube than on a mesh.
- Any problem of size W will run more efficiently on a p-processor hypercube than on a p-processor 2-D mesh (both with SF routing).
- It is costlier to build a parallel computer with 500 processors of 1 MFLOPS each than to construct a serial computer with a single 500 MFLOPS processor.

4. Consider the multiplication of an $n \times n$ matrix with an $n \times 1$ vector on a $\sqrt{p} \times \sqrt{p}$ mesh with CT routing.

   - What is the parallel run time for $p = n^2$ if the checkerboard mapping as shown in Figure 5.9 of the text is used?
   - What is the parallel runtime for checkerboard mapping with $p = n$? Is the run time asymptotically different from the case when $p = n^2$?

- What is the parallel run time for $p = n$ if the striped mapping as shown in Figure 5.8 of the text is used?

5. Solve Problems 4.8, 4.9, and 4.10 from the textbook.

6. A drawback of increasing the number of processors without increasing the total work load is that the speedup does not increase linearly with the number of processors, and the efficiency drops monotonically. Based on your experience with Problem 5 above, discuss whether using scaled speedup instead of standard speedup solves the problem in general.

CSCI-5151 Winter 1992

HW # p1 (in lieu of Project # 1; due on Feb 18, '92)

1. n distinct keys are stored in a MIMD parallel machine with p processors. Initially each processor contains n/p keys (n is a multiple of p). Assume a number k is available at a corner processor (i.e., node 0 in hypercube and node (0, 0) in 2D mesh). The problem of finding out which processor containing k can be done in the following steps:

   1) Broadcast k to every processor.

   2) Each processor matches k with its own keys.

   3) Each processor sends back (1, processor_id) if k is one of its own keys, and (0, -1) otherwise.

   At the end of computation, the corner processor gets the id of the processor which contains k. Answer the following questions on a SQRT(p) x SQRT(p) mesh and a heypercube of dimension log(p) respectively.

   a) Describe the algorithm in each step. Your algorithm should be as efficient as possible.

   b) Compute parallel time, speedup and efficiency in terms of p, n, t_c, t_w and t_s, where t_c is the comparison time, t_w is the time for sending one value from a processor to one of its nearest neighbors, and t_s is the message startup time.

   Do parts c) and d) for t_c = 1, t_w = 1 and t_s = 100.

   c) If you want to get 0.5 efficiency in 64 processors, what should n be?

d) If you want to get efficiency E in p processors, what should n be as a function of p and E?

2. Given n = 2^ (k/2) for a positive even integer k, write a function that maps elements of a k-dimensional hypercube into nodes of an n X n wraparound mesh so that all mesh connections map directly onto connections in the hypercube. The function should take as input a processor number of the hypercube and should yield as output two integers i and j (0 <= i <= n, 0 <= j <= n) which are the coordinates of the location of the processor in the corresponding mesh. For example, in the mapping of a 16-processor mesh to a hypercube shown in the figure in HW # 1 handout, given the value 7, the function should yield the values 1,2.

3. (a) Describe an algorithm for multiplying two n X n matrices using n^2 processors on an EREW PRAM multiprocessor and derive an expression for its Parallel Run Time. Is your algorithm cost optimal ?

   (b) How can the algorithm of part (b) be modified to work on p processors such that p < n^2 and n is divisible by Sqrt(p). Derive expressions for Parallel Run Time, Efficiency and the Iso-efficiency function in terms of p and n. (only order terms required.)

4. Briefly describe an algorithm to multiply a "n" X "n" matrix with a "n" X "1" vector on a SQRT(p) X SQRT(p) processor mesh. Assume that the input vector is distributed horizontally in the first row of the mesh processors and the resultant vector is also expected to lie in the first row of processors. Derive expressions for Parallel Run Time, Speedup, Efficiency and Iso-efficiency Function in terms of "n" and "p". (only order terms required.)

CSCI 5151 Winter - 1992

HW # P2 (A possible substitute for project # 2)

Due Date - Thursday, March 12, 1992.

1. What is cut-through (or worm-hole) routing ? Given an example of an algorithm that will run faster on a mesh with cut-through routing than on a simple mesh.

2. State whether the following statements are true or false. Provide a brief justification AND an example in support of each of your answers. No credit will be given without a correct explanation.

(i) It is possible to obtain any efficiency between 0 and 1 for an algorithm architecture combination for which the iso-efficiency function exists.

(ii) Among two parallel algorithms with asymptotically different iso-efficiency functions, the one with the asymptotically smaller iso-efficiency function will always achieve a higher efficiency than the other algorithm for a given problem size and a given number of processors.

(iii) The iso-efficiency functions for a parallel algorithm architecture combination for maintaining two different efficiencies E1 and E2 ($0 < E1 < 1$ and $0 < E2 < 1$) can be asymptotically different from each other.

3. Consider the implementation of the simple matrix multiplication algorithm described in section 4.1 of the report "Scalability of Prallel Algorithms for Matrix Multiplication" on a mesh connected multicomputer.

1) What are the parallel time, speedup and efficiency in terms of n, p, t_c, t_w and t_s if two nXn matrices are being multiplied using p processors ?

2) Derive an expression for its isoefficiency function if $t_s = 0$.

3) What is the rate at which the problem size is required to grow (as a function of p only) for $t_c = 1$, $t_w = 1$ and $t_s = 0$ if an efficiency of 0.25 needs to be maintained ? What is this rate to maintain an efficiency of 0.75 ?

4) What is the parallel run time, speedup, and efficiency for $n = 256$, $p = 64$, $t_c = 1$, $t_w = 1$, $t_s = 0$? In the following parts you would be required to recompute these values, changing one parameter at a time.

5) What are the values of parallel run time and efficiency if the number of processors is increased 16 times (i.e. to 1024) in part 3)? By what factor should the problem size ($W = n^3$) increase in order to maintain the same efficiency as obtained in part 3)?

6) Suppose the processor speed in part 3) is increased 16 times (i.e. new $t_c = 1/16$). What are the values of parallel run time and efficiency now ? By what factor should the problem size change to maintain the same efficiency as in part 3) ?

7) Suppose the communication bandwidth is now increased by a factor of 16 (i.e. new $t_w = 1/16$). What are the new parallel run time and efficiency ? By what factor should the problem size change to maintain the same efficiency as in part 3) ?

8) For this algorithm, what would you say in general about the dependency of efficiency on processor power, network bandwidth, the number of processors and problem size?

4. Give parallel run time, speedup and efficiency for the algorithms described in sections 4.2, 4.4.1, and 4.4.2 of the report in terms of t_s, t_w, and t_c for hypercube. Which of these algorithms would you use for the following values of n and p ? Why ?

1) $n = 2000$; $p = 1,000,000$
2) $n = 1000$; $p = 2,000,000$
3) $n = 1000$; $p = 4,000,000$
4) $n = 3000$; $p = 4,000,000$
5) $n = 3000$; $p = 10,000,000$

The 4.4.2 algorithm performs better than 4.4.1 algorithm for certain combination of n and p even though it has worse scalability. What feature/features of these algorithms is/are responsible for this ?

5. Consider matrix multiplication on a linear array of processors. Assume n X n matrices being multiplied on a linear array of n processors, such that each processor has one column of each of the matrices to start with and is expected to generate the corresponding column of the result matrix.

1) Describe an algorithm for performing this with the initial data distribution as given above. What is the parallel execution time ? Is the algorithm cost optimal ?

2) Modify the above algorithm to work for $p < n$ by assigning n/p columns of the matrices to each processor. Derive expressions for parallel execution time, speedup and efficiency in terms of n, p, t_c, t_w and t_s.

3) For what values of p (in terms of n) will the algorithm be cost optimal ?

4) Assuming $t_s = 0$, derive an expression for iso-efficiency in terms of p, t_c, t_w and E.

5) How should the problem size grow (as a function of p) to maintain an efficiency of 0.25 for $t_c = 1$, $t_w = 1$ and $t_s = 0$?

6. Consider multiplication of n X n matrices on a linear array of processors with $p = n^2$, such that each processor has exactly one element of the source matrices.

1) Describe a mapping of the matrix elements onto the processors of the linear array.

2) Describe an algorithm to perform matrix multiplication with the initial data distribution as in part 1) above. What is its parallel execution time ? Is the algorithm cost optimal ?

3) Now modify the data mapping and the algorithm to work on $p < n^2$. Derive expressions for parallel execution time, speedup and efficiency

in terms of n, p, t_c, t_w and t_s.

4) For what values of p (in terms of n) will the algorithm be cost optimal ?

5) Assuming t_s = 0, derive an expression for iso-efficiency in terms of p, t_c, t_w and E.

FINAL EXAMINATION

90 points / 2 Hours

NOTE: When deriving expressions for runtime, you may like to detail all the appropriate steps. Otherwise, no partial credit will be awarded for incorrect expressions

1. (5×2 **points**)

   State whether the following statements are true or false (no explanation required):

   - Bus-based parallel computers (*i.e.,* parallel computers that use a global bus to connect memory to all the processors) are scalable in terms of cost and performance.

   - Processors are synchronized at instruction level in an MIMD computer.

   - Shared address space architectures can easily emulate a message passing paradigm.

   - It is costlier to connect $4096 (= 2^{12})$ processors in a hypercube network than a three-dimensional mesh.

   - The completely-connected network performs at least as fast as any other static interconnection network for any application.

2. $(4 \times 4$ **points)**
State whether the following statements are true or false. Give a brief explanation (2-3 lines) supporting your answer.

- SIMD parallel computers are more flexible to program than MIMD parallel computers.

- Processors in a SIMD computer are cheaper than in MIMD computers.

- Given a sequential program, it is easier to convert it to a parallel program for a message passing computer (than for a shared address space computer).

- Data-parallelism can be exploited only on SIMD architectures.

3. **1 + 1 + 1 + 2 points**
The sequential runtime of an algorithm is given by $W = n \log n$. The corresponding parallel runtime is given by $T_P = np + n \log p$.

- What is the processor-time product for this algorithm?

- Is this algorithm cost optimal in the following cases:

  $p = n$

  $p = \sqrt{n}$

  $p \leq \log n$

4. **8 points**

   Consider all-to-all broadcast on a hypercube, a mesh and a linear array, each with cut-through routing. Assume that the startup time $t_s = 0$. Furthermore, assume that the cost of a parallel computer is given by the total number of wires required by the interconnection network (Page 38 of text). Which of these architectures is most cost effective?

   Hint: Show that for a given time for all-to-all broadcast, one of the architectures is cheaper than the other.

5. **5 + 5 points**

(a) Consider Gaussian Elimination of an $n \times n$ system of equations (without pivoting) on a p processor hypercube with cut-through routing. Is one of the following mappings better than the other, or do they have the same performance: (i) Block-Checkerboard mapping; and (ii) Cyclic-Checkerboard.

(b) Repeat Problem 5(a) for multiplying two $n \times n$ matrices (instead of Gaussian Elimination).

6. **13 points**

Consider the problem of multiplying two $n \times n$ matrices A and B on an n processor hypercube with cut-through routing. Each processor is assigned a single row of matrix A and a single row of matrix B. Briefly describe the algorithm. You can use a combination of figures and text to describe the algorithm. Compute the parallel runtime of this algorithm in order terms.

7. **13 points**

Consider the parallel formulation of quicksort for a hypercube given in Section 6.4 of the text. Assume the worst case pivot selection. This means that after the first step, every element from one subcube is communicated to the other subcube (since every element is smaller than the pivot). Now, each processor in the second subcube has $2n/p$ elements. Similarly, after the second step, all elements end up on a fourth of the hypercube. Each processor now has $4n/p$ elements now.

Compute the parallel runtime of this algorithm. You can assume that $t_s = t_h = 0$.

8. **7.5 + 7.5 points**

(a) A key component of the parallel formulation of Prim's algorithm is the partitioning of the adjacency matrix onto processors. Consider a parallel formulation of Prim's algorithm for finding the MST for a graph with n nodes. We are given a hypercube connected computer with cut-through routing and n^2 processors. Each element in the adjacency matrix is assigned to a single processor. Compute the parallel runtime of this algorithm.

(b) Now, consider a p processor hypercube with cut-through routing. The adjacency matrix is partitioned among the processors using a block-checkerboard mapping. What is the parallel runtime of this algorithm.

CSci 5299 Winter 91

Final Exam March 18, 1991

1\. (a) Which of the following statements is FALSE for a MIMD computer ?

i) PEs are synchronized at instruction level.

ii) Each PE must have a Control Unit.

iii) Is costlier to build than a SIMD computer with the same number of processors (of comparable speed).

iv) PEs communicate through an interconnection network.

(b) Which of the following statements is FALSE ?

i) The network providing access to shared memory is usually complex and a large fast network can be very expensive.

ii) A message passing architecture can easily emulate a shared memory paradigm, but the reverse is not true.

iii) Using caches can help in reducing the load on the network providing access to shared memory.

iv) Shared memory architectures readily support rapid access by all the processors to large dynamically changing data structures.

(c) In which of the following networks, the Maximum Bandwidth per processor decreases as the number of processors in the configuration is increased ?

i) Crossbar.

ii) Bus.

iii) Hypercube.

iv) Multistage.

(d) In which of the following networks does the total wire length (used in the interconnection network) per processor remain constant ?

i) Hypercube.

ii) Multistage.

iii) A wrap around mesh.

iv) A 3-D mesh.

(e) Which of the following statements is TRUE ?

i) It is possible that a cost optimal parallel algorithm may not have an iso-efficiency function.

ii) It is possible that a parallel algorithm having an iso-efficiency function may not be cost optimal.

iii) It might be impossible to obtain a particular efficiency E E <= 1) on a parallel algorithm for which an iso-efficiency function exists.

iv) Among two parallel algorithms with asymptotically different o-efficiency functions, the one with the asymptotically smaller iso-efficiency function will always achieve a higher efficiency than the other algorithm for the same problem size and the number of processors.

(f) Which of the following statements is FALSE ?

i) Any parallel algorithm will have an asymptotically better (smaller) o-efficiency function on a hypercube than on a mesh.

ii) For a given problem size and the number of processors, a parallel gorithm can not yield a higher efficiency on a mesh than on a hypercube with identical processor speed and channel bandwidth.

iii) If a parallel algorithm is cost optimal on a tree connected chitecture, then it has to be cost optimal on a hypercube.

iv) The iso-efficiency functions for a parallel algorithm architecture mbination for maintaining two different efficiencies E1 and E2 (0 < E1 < 1 and 0 < E2 < 1) can be asymptotically different from each other.

2. Which of the following statements are TRUE ? Give a one line stification for your answer, whether the statement is true or false. Credit will be given only if the justification is correct.

(a) If enough effort and money is spent, it is possible to construct arbitrarily fast sequential processor.

(b) A parallel machine with 100 processors of 100 MIPS each will liver ten times the throughput of a machine with a single 1000 MIPS processor on an application.

(c) It is usually easier to convert a sequential algorithm to a parallel gorithm for a shared memory architecture than for a message passing architecture.

(d) Commercially available shared memory computers often allow each ocessor some local memory also which is directly accessible only from its parent processor.

(e) The main challenge of parallel processing is to build massively rallel computers rather than to exploit their computational power.

3. Briefly describe an efficient algorithm for finding All Sums a hypercube and derive expressions for Parallel Execution Time, Speedup, Efficiency and Iso-efficiency Function in terms of "n" and "p", ere "n" is the total number of data items and "p" is the number of processors to be used.

4. Briefly describe an implementation of the checkerboard version of oyd's algorithm for finding the shortest paths on a linear array of processors. If an operation of the type a = min(b, c+d) takes time t_c, the communication startup time is t_s and the channel bandwidth is 1/t_b words per second, then derive exact expressions for Parallel Run Time and Efficiency. Derive an asymptotic expression for the Iso-efficiency function.

5. (a) Describe an algorithm for matrix multiplication that has a linear o-efficiency function. You can chose the architecture from any interconnection networks described in the class or from any of the shared memory models.

(b) Describe an algorithm for multiplying two n X n matrices using 2 processors on an EREW PRAM multiprocessor and derive an expression for its Parallel Run Time. Is your algorithm cost optimal ?

(c) How can the algorithm of part (b) be modified to work on p ocessors such that p < n^2 and n is divisible by Sqrt(p). Derive expressions for Parallel Run Time, Efficiency and the Iso-efficiency function in terms of p and n. (only order terms required.)

CSci 5152 - Spring 1994
Final Exam

Last Name : ____________________, First Name : ____________________.

ID # : ____________________

	Maximum Points	Points Awarded
Q. 1	22	
Q. 2	20	
Q. 3	14	
Q. 4	24	
Q. 5	20	
Total	100	

Name : ______________________

Q. 1 :

($7 + 8 + 2 + 5 = 22$ *points*)

Briefly describe an implementation of the block-checkerboard version of Floyd's algorithm for finding the shortest paths on a em linear array of processors (without pipelining). If an operation of the type $a = \min(b, c + d)$ takes time t_c, the communication startup time is t_s and the channel bandwidth is $1/t_w$ words per second, then derive exact expressions for Parallel Run Time and Efficiency. Derive an asymptotic expression for the Iso-efficiency function.

Name : ____________________

Name : ____________________

Q. 2 :

$(10 + 10 = 20\ points)$

In the analysis of various load-balancing schemes presented in Section 8.4.2, the cost of transferring work is assumed to be independent of the amount of work transferred. Consider the scenario in which the work-transfer cost is a function of the amount of the work transferred and the stack size for representing a search space of w nodes varies as a function $f(w)$ of the number of nodes.

(a) If the GRR load-balancing scheme is used, what is the total communication overhead and the isoefficiency function on a hypercube for $f(w) = \sqrt{w}$?

(b) What is the communication overhead and the isoefficiency function of GRR-M on a network of workstations when $f(w) = \log w$?

Name : _______________________

Q. 3 :

$(4 + 6 + 4 = 14$ *points*)

Consider the parallel algorithm for the 0/1 knapsack problem on a hypercube in Section 9.1.2.

(a) What is the smallest value of c for which the algorithm is cost-optimal on a p-processor hypercube?

(b) Derive expressions for speedup and efficiency.

(c) Does the efficiency of this algorithm approach 1.0 as the problem size is increased for a fixed number of processors? Give a very brief explanation for your answer.

Name : ____________________

Q. 4 :

$(2 + 7 + 7 + 8 = 24$ *points*$)$

You are given an algorithm for which the serial execution time $W = t_c \times n \log n$. When implemented on a p processor hypercube, the parallel execution time T_p is $\frac{t_c \times n \log n}{p} + \frac{t_w \times n \log n}{p} + \frac{t_h \times n \log p}{p}$. Assume that $t_c = 1, t_w = 0.2, t_h = 1$.

a. Write expressions for speedup and efficiency.

b. What is the iso-efficiency function if an efficiency of 0.6 is desired ?

c. How will the iso-efficiency function change (if at all) if an efficiency of 0.4 is desired ?

d. Repeat parts **b** and **c** above when $t_w = 1$ and everything else is the same.

Name : ____________________

Q. 5 :

$(10 + 10$ *points*$)$

Consider an $n \times n$ banded unstructured sparse matrix with m nonzeros per row and a bandwidth w. Derive expressions for the run time and isoefficiency function when

(a) $w = 100$.

(a) $w = \sqrt{n}$.

Name : ____________________

CSci 5151 - Winter-1992
Final Exam

Last Name : ____________________, First Name : ____________________.

ID # : ____________________

	Maximum Points	Points Awarded
Q. 1	12	
Q. 2	12	
Q. 3	8	
Q. 4	16	
Q. 5	16	
Q. 6	12	
Q. 7	12	
Q. 8	12	
Total	100	

Name : ____________________

Q. 1 :

($3 \times 4 = 12$ *points*)

a. Which one of the following iso-efficiency relations between problem size W and the number of processors p is NOT realizable ? Why ?

b. Which one of the following speedup curves IS realizable in practice ? W hy ?

Name : ______________________

c. You are given two different algorithms A_1 and A_2 to solve the same problem on a given architecture. It is known that for any given values of W and p, A_1 yields a higher efficiency than A_2. What factor(s), if any, will force you to use A_2 instead of A_1 to solve this problem ?

d. You have to implement a parallel algorithm on two different parallel machines M_1 and M_2 with different interconnection networks. It is known that for the given algorithm, M_1 always yields a higher efficiency than M_2 for any given values of W and p. What factor(s), if any, will force you to use M_2 instead of M_1 to implement this algorithm ?

Q. 2 :

($3 \times 4 = 12$ *points*)

State whether the following statements are true or false. Provide a brief 1 or 2 lines explanation for your answer. Credit will be given only if the explanation is correct. For part 3, use the following definition of cross section bandwidth :

If an interconnection network connecting p processors is partitioned into two equal halves, each containing $\frac{p}{2}$ processors, then the cross section bandwidth $CW(p)$ of the architecture is defined as the total number of communication channels crossing the partition.

Name : ____________________

a. It takes $O(\log p)$ time to synchronize all the processors on a p - processor SIMD hypercube.

b. It is costlier to build a parallel computer with 1000 processors of 10 MFLOPS each than to construct a serial computer with a single 10000 MFLOPS processor.

c. $CW_{hypercube}(p) = \sqrt{CW_{fully\ interconnected\ network}(p)}$.

d. The best iso-efficiency function that any matrix multiplication algorithm can achieve on any parallel architecture is $O(p^{1.5})$.

Name : ____________________

Q. 3 :

(8 *points*)

Consider the problem of performing matrix - vector multiplication on a hypercube. Assume that you are given a hypercube with $n^2 = 2^{2q}$ processors to multiply an $n \times n$ matrix with a vector of length n. Initially, the vector is distributed among the same n processors that have the first column of the matrix.

Before performing the multiplication, the vector has to be distributed such that each row of processors has the vector. The figures below show the elements of the vector before and after the distribution step.

Briefly describe the data communication steps that are required for the above so that each processor has all the data it needs to start the computation. You may use pictures to augment textual description.

How much time does it take (in order terms) to perform this initial data distribution step ?

Name : ______________________

Q. 4 :

(16 *points*)

You are given an algorithm for which the serial execution time $W = t_c \times n \log n$. When implemented on a p processor hypercube, the parallel execution time T_p is $\frac{t_c \times n \log n}{p} + \frac{t_w \times n \log n}{p} + \frac{t_h \times n \log p}{p}$. Assume that $t_c = 1, t_w = 0.2, t_h = 1$.

a. Write expressions for speedup and efficiency.

b. What is the iso-efficiency function if an efficiency of 0.6 is desired ?

c. How will the iso-efficiency function change (if at all) if an efficiency of 0.4 is desired ?

d. Repeat parts **b** and **c** above when $t_w = 1$ and everything else is the same.

Name : ______________________

Name : ____________________

Q. 5 :

(16 *points*)

In section 4.4 of Akl's book (page 94), a sorting algorithm (CRCW SORT) is given that sorts n numbers on a n^2 processor CRCW PRAM in O(1) time. In step 1 of this algorithm, the ranking of each element is computed.

Implement this step on a mesh connected parallel computer with n^2 processors. Initially these n numbers are stored in the first row of processors such that each processor has one element. At the end of this computation, the first row of processors is required to get the rank of the number it initially had. For example, if (54, 23, 76, 10) are stored in the first row of a 4×4 mesh, then at the end of the computation, the first row of processors should have (2,1,3,0).

Describe an algorithm to perform this step. you can use figures to augment textual description. What is the parallel execution time of this step (in order terms) ?

Name : ____________________

Name : ____________________

Q. 6 :

(12 *points*)

Consider an array of n numbers which are distributed on an n processor parallel computer in such a way that each processor has exactly one number. We are interested in the following three operations:

1. All to all broadcast. (every processor needs to get the entire array)
2. One to all broadcast. (processor 0 needs to broadcast its number to all other processors)
3. Sumscan.

What is the best parallel execution time (in order terms) of each of these operations on a linear array of processors, a mesh and a hypercube, with n processors on each machine. Fill in the following table with your values.

	Linear Array	Mesh	Hypercube
All to all broadcast			
One to all broadcast			
Sumscan			

Q. 7 :

($6 + 6 = 12$ *points*)

Consider the following parallel all-pairs shortest path algorithms :

Dijkstra's source-parallel (DSP)

Floyd's striped (FS)

Floyd's checkerboard (FC)

Floyd's pipelined checkerboard (FPC)

Cube shortest path (CSP)

(The first four algorithms are described in "Scalability of Parallel Algori thms for the All-Pairs Shortest-Path Problem" by Kumar and Singh in your reading list. The CSP algorithm is described in Akl's book, section 10.5, page 259.)

a. In the following table, indicate which algorithm works in least parallel run time (in order terms only) on each of the following architectures for finding shortest paths between all pairs of a n - node graph. Assume that an unlimited number of processors are available.

Name : ____________________

Architecture	Fastest Algorithm	Parallel Execution Time
Mesh		
Hypercube		
CREW PRAM		

b. Now consider only the cost-optimal algorithms and fill the following table accordingly. Note that some algorithms are cost optimal for a certain combination of n and p only. For these algorithms, use only as many processo rs as can be used cost-optimally.

Architecture	Fastest Algorithm	Parallel Execution Time
Mesh		
Hypercube		
CREW PRAM		

Q. 8 :

(12 *points*)

Bitonic sort maps naturally on a hypercube as described in Fox's book (page 334), and takes $O((\log n)^2)$ time with n processors to sort n numbers. Consider an implementation of this algorithm on an n processor mesh (without wrap-around connections). This can be done by numbering the mesh processors appropriately, and then runnin g the algorithm similar to that in Fox's book.

For a 4×4 mesh, number the mesh processors appropriately from 0000 to 1 111 such that an algorithm similar to that in Fox's book can be executed directly.

Derive an expression (in order terms only) for the parallel execution time of this algorithm on an $\sqrt{n} \times \sqrt{n}$ mesh.

Name : ______________________

CSci 5151 Winter '92

NCUBE Project # Due date : Feb. 18, 1992

Develop host and node programs on the Ncube with the following specifications :

1. The host program takes 3 integers "d", "s" and "i" as inputs such that 0 <= d <= 6; 0 <= s <= 2^d - 1 and "i" is any integer.

2. The host program allocates a "d" dimensional cube and loads the node program in all the nodes.

3. The host program sends the integer value "s" to all the nodes.

4. The host program sends the integer value "i" to the node numbered "s".

5. Now the node whose index is "s" becomes the source and initiates the broadcast of the data value "i" to all the other nodes in the cube. Note that no node should perform more than "d" communication steps - sends and receives combined.

6. Each node adds its own index (node number) to the received data i.e. "i".

7. The host program collects the values thus calculated by each node prints them.

8. Your program should be such that each communication occurs between immediately neighboring processors only, and if it takes time "t" to pass one message between two neighboring processors, then the total time that the program spends in message passing among the nodes should not exceed "d X t". This does not include the messages exchanged between the nodes and the host.

For example if d = 3, s = 5, i = 10, then the host program allocates a 8 node cube and sends the integer value 10 to node no. 5. Now this value is broadcast to all the nodes in the cube. The total number of nreads and nwrites combined for any node should not exceed 3.

(Note that these 3 do not include the nreads and nwrites for communication with the host.) Finally the following results should be printed by the host :

```
Node # 0  ---  10
Node # 1  ---  11
        .
        .
        .
Node # 7  ---  17
```

The NCUBE can not handle more than 3 users at a time. So please run the "xnodes" command before running your program and if more than three processes are running on the machine, wait till someone gets off. Otherwise the machine will crash and every one will have to wait till someone from the systems staff reboots it.

Unless the machine is down continuously for a long stretch, (like a couple of weeks) non-availability of the machine will NOT be considered as a valid reason for not meeting the submission deadline. The deadline is final and will NOT be relaxed for anyone under any circumstances.

Parallel Programming Project
CSci 5152 (Spring 1994)
Due Date: April 26 for Part I, May 10 for Part II, May 24 for Part III
Late submission penalty: 10% per day

This nCUBE programming project has three parts. For each part, you must

- provide well documented source code by the due date.
- submit all results and graphs etc. by the due date.
- provide your login name and the location of executable and data files on brutus.
- make sure that all output is on stdout or stderr.
- make sure that all executable and data files have read and execute permission for all users.
- do all your coding in C using either the nCUBE message passing primitives or PPL.

1 Part I

Implement dense matrix vector multiplication for $n \times n$ matrices with checkerboard partitioning. Provide a table of run times on 1, 4, and 16 processors for $n = 32$, 64, 128, and 256. Plot speedup curves for each value of n on the same graph.

All matrices and vectors should have integer entries. Generate the input matrix and the vector randomly such that each entry is between 0 and 10. For the case of $n = 32$, print the product vector for $p = 1$, 4, and 16 and make sure that the result matches for each value of p.

2 Part II

Implement Prim's algorithm for finding a minimum spanning tree of an n-node graph. Provide a table of run times on 1, 2, 4, 8, and 16 processors for $n = 32$, 64, 128, and 256. Plot speedup curves for each value of n.

The adjacency matrix of the graph should be a symmetric randomly generated integer matrix with entries between 0 and 100 (generate the lower triangular matrix randomly, and copy the same in the upper triangular part to make sure that it is symmetric). Moreover, regard each entry greater than 50 as infinity; i.e., regard the nodes in question as disconnected. For the case of $n = 32$, print the resulting tree for $p = 1$, 2, 4, 8, and 16 and make sure that the result matches for each value of p.

3 Part III

Implement bitonic sort of a randomly generated sequence of n numbers. Provide a table of run times on 1, 2, 4, 8, and 16 processors for n = 64, 256, 1024, and 4096. For the case of n = 64, print the initial and the sorted sequences for p = 1, 2, 4, 8, and 16 and make sure that the result matches for each value of p.

Parallel Programming Project
CSci 5152 (Spring 1994)
Due Date: May 3
Late submission penalty: 10% per day

This nCUBE programming project has two parts. For each part, you must

- provide well documented source code by the due date.
- submit all results and graphs etc. by the due date.
- provide your login name and the location of executable and data files on brutus.
- make sure that all output is on stdout or stderr.
- make sure that all executable and data files have read and execute permission for all users (also make sure that the source files do not have read permission for other users).
- do all your coding in C using either the nCUBE message passing primitives or PPL.

4 Part I

Implement dense matrix vector multiplication for $n \times n$ matrices with checkerboard partitioning. Provide a table of run times on 1, 4, and 16 processors for $n = 32$, 64, 128, and 256. Plot speedup curves for each value of n on the same graph.

All matrices and vectors should have integer entries. Generate the input matrix and the vector randomly such that each entry is between 0 and 10. For the case of $n = 32$, print the product vector for $p = 1$, 4, and 16 and make sure that the result matches for each value of p.

Write an expression for the parallel run time in terms of n, p, t_c, t_s, and t_w, where t_c is the time for an integer multiplication-addition pair. The serial run time is $t_c n^2$. Compute the value of t_c using the run time for $n = 256$ and $p = 1$. Using this value of t_c and the run times for $n = 256, p = 16$ and $n = 128, p = 16$, compute the values of t_s and t_w.

5 Part II

Implement Prim's algorithm for finding a minimum spanning tree of an n-node graph. Provide a table of run times on 1, 2, 4, 8, and 16 processors for $n = 32$, 64, 128, and 256. Plot speedup curves for each value of n.

The adjacency matrix of the graph should be a symmetric randomly generated integer matrix with entries between 0 and 100 (generate the lower triangular matrix randomly, and copy the same in the upper triangular part to make sure that it is symmetric). Moreover, regard each entry greater than 50 as infinity; i.e., regard the nodes in question as disconnected. For the case of $n = 32$, print the resulting tree for $p = 1$, 2, 4, 8, and 16 and make sure that the result matches for each value of p.