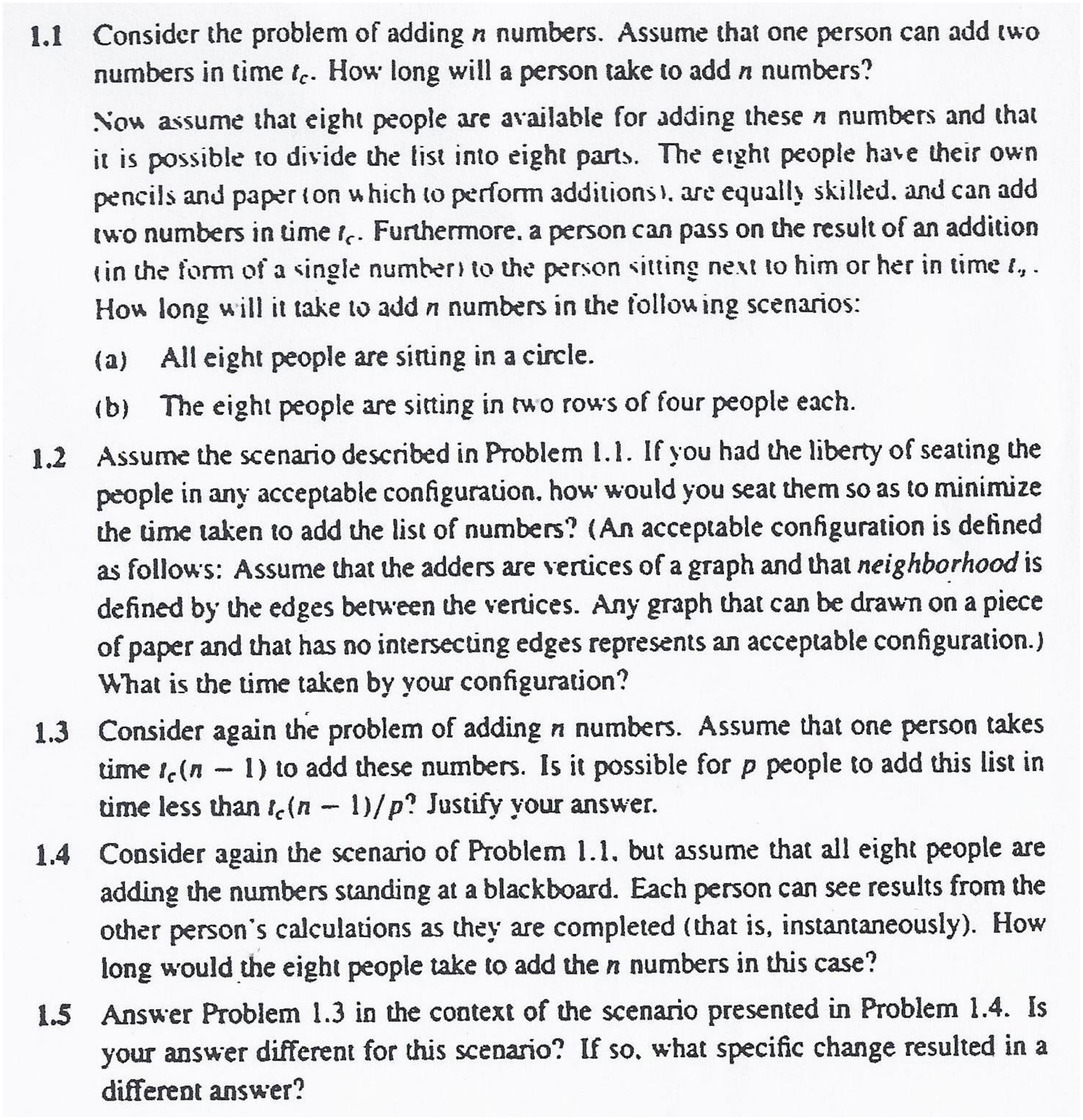
Assignment 1: Models, Algorithms, and Parallelism CSci6702 Parallel Computing

Answer the following questions. Print or type neatly your work. Draw diagrams where they are useful. Justify your answers. Where necessary write down your assumptions.



1.1 Consider the problem of adding n numbers. Assume that one person can add two numbers in time t_c. How long will a person take to add n numbers?

Now assume that eight people are available for adding these n numbers and that it is possible to divide the list into eight parts. The eight people have their own pencils and paper (on which to perform additions), are equally skilled, and can add two numbers in time t_c. Furthermore, a person can pass on the result of an addition (in the form of a single number) to the person sitting next to him or her in time t_w. How long will it take to add n numbers in the following scenarios:

(a) All eight people are sitting in a circle.

(b) The eight people are sitting in two rows of four people each.

1.2 Assume the scenario described in Problem 1.1. If you had the liberty of seating the people in any acceptable configuration, how would you seat them so as to minimize the time taken to add the list of numbers? (An acceptable configuration is defined as follows: Assume that the adders are vertices of a graph and that *neighborhood* is defined by the edges between the vertices. Any graph that can be drawn on a piece of paper and that has no intersecting edges represents an acceptable configuration.) What is the time taken by your configuration?

1.3 Consider again the problem of adding n numbers. Assume that one person takes time $t_c(n-1)$ to add these numbers. Is it possible for p people to add this list in time less than $t_c(n-1)/p$? Justify your answer.

1.4 Consider again the scenario of Problem 1.1, but assume that all eight people are adding the numbers standing at a blackboard. Each person can see results from the other person's calculations as they are completed (that is, instantaneously). How long would the eight people take to add the n numbers in this case?

1.5 Answer Problem 1.3 in the context of the scenario presented in Problem 1.4. Is your answer different for this scenario? If so, what specific change resulted in a different answer?

1.6 For each regular architecture give the following parameters:

Architecture	**# of processors**	**# of wires**	**Diameter**	**Bisection bandwidth**	**Network degree**
Linear Array	p				
Ring	p				
2D Mesh	p				
3D Mesh	p				
Hypercube	p				

1.7 A p-processor fine grained model assumes that each processor can store only a constant number of data items and can communicate with all of its neighbors in O(1) time. For each of the architectures listed in Question 1.6 describe an algorithm for computing the largest (ie. Max) value. Input: each processor stores a single value. Output: A designated processor P_i stores the largest of the input values. For each algorithm compute the speed-up. In which cases is the speed-up optimal? Be careful to define what you mean by optimal. How do the architectural parameters impact the performance of your algorithms?

1.8 Mary has implemented a parallel sorting algorithm and run with 1024K randomly selected items varying the number of processors (see table below). She also ran an optimal sequential version which took 1066 seconds to run on a single node of her machine.

Processors	1	2	4	8	16	32	64	128
Time in seconds	1404	731	348	202	119	93	84	90

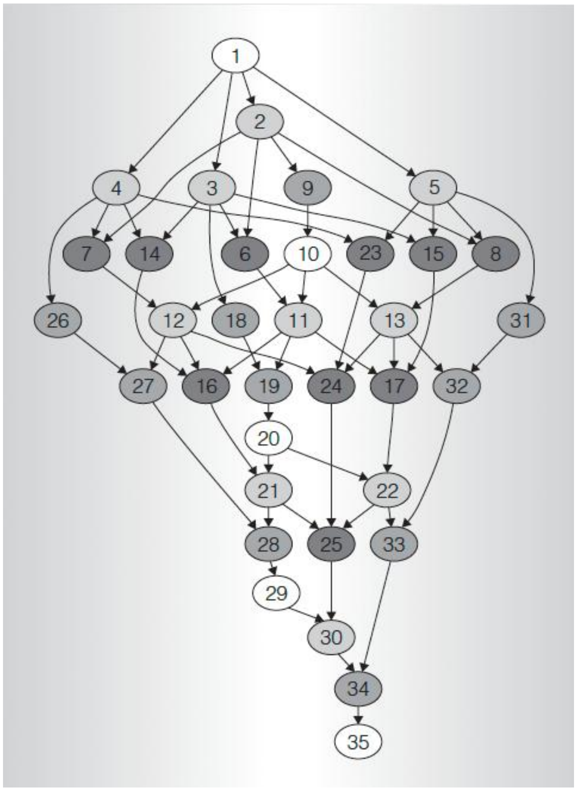
a) Graph time on the y-axis and processors on the x-axis. Be sure to title the graph and list any other important parameters.
b) Graph the corresponding speed-up and relative speed-up. Be sure to add a "linear" speed-up line for comparison.
c) Graph the parallel efficiency.
d) What might be happening to account for the shape of the graphs? Describe in your own words what you think might be happening. Is her code good?

1.9 Suppose that a program obeys Amdahl's law with a sequential fraction f = 5% and the execution time with one processor is 10 seconds. Graph time vs. processors, and speed-up, and efficiency for 1 <= p <= 100.

1.10 Answer the following question:

Assume that the size of a problem obeying Amdahl's law is increased as the number, P, of processors increases. The absolute amount of sequential work is a constant, S (in seconds), but the amount of Parallel work, Q, increases with problem size, N, as $Q = qN$. The problem size scales with the number of processes as $N = cP$.
For $P = 10$ processors, an appropriate size problem has $N = 1{,}000$ and a sequential fraction of 5%. Graph the speedup and efficiency from $P = 10$ to $P = 100$ in steps of 10 processes.

1.11 Consider the following task graph where the integer associated with each node indicates the task id and the colour indicates the number of time units it takes to execute the it (white = 1, light grey = 3, medium grey = 5, dark grey = 9).



(Image from http://doi.ieeecomputersociety.org/cms/Computer.org/dl/mags/mi/2010/05/figures/mmi2010050016a.gif)

a) What is the **span** and the **work** for this task graph?
b) What is the maximum speed-up?
c) If a parallel computer had 3 processors and a "perfect" or "optimal" scheduler how long would it take to execute this task graph assuming a scheduling overhead per job of one time unit? Draw a diagram of the schedule.

Bonus Question

Describe and analyse an efficient fine-grained algorithm that sorts *n* distinct elements on stored on the first *n* processors of a hypercube of size *p = n^2*. How are you going to use all those processors?
Hints: 1) Find the rank of each item and then move them to that position. 2) Think of the hypercube as an *n * n* mesh where each row and column is a hypercube. 3) The following operations take time O(log p) on a hypercube: min, max, broadcast, sum, partial-sum.