These questions are examples of the type of questions used in the final exam for CS6230 the question paper usually consists of three questions. in which you will be expected to attempt to answer the equivalent of two questions in an hour and thirty minutes.

SCHOOL OF COMPUTING
UNIVERSITY OF UTAH
CS6230 FINAL
Time allowed: 1 hour and Thirty Minutes
Attempt TWO questions.

Question 1

(a) Suppose that an operation with 2^p elements may be recursively decomposed into two operations with 2^{p-1} elements operations. Show with the aid of a diagram how this can be recursively decomposed into p stages until $p = 1$. [2 marks]

(b) Explain how you would implement such an operation in parallel, assuming that the decomposition at the pth stage involves the communication of 2^{p-1} values between pairs of adjacent processes [2 marks]

(c) Perform a speed-up and efficiency analysis for this case. [4 marks]

(e) The Quicksort algorithm has a serial complexity of $O(nlog(n))$ on average and works by using pivots to recursively split lists. Explain by using an illustrative example how you would implement Quicksort in parallel using successively more processors. [3 marks]

(f) Show that if the list of n numbers is split recursively and evenly then

$$t_{comp} = n + n/2 + n/4 + n/8....$$

and that

$$t_{comm} = (t_{startup} + n/2t_{data}) + (t_{startup} + n/4t_{data}) + (t_{startup} + n/8t_{data}) + etc$$

[4 marks]

(d) Hence calculate the speedup and isoefficiency of the parallel algorithm and explain how the performance might differ in reality. [3 marks]

(d) Explain the advantages of a workpool approach in this context and modify your model to include this. [2 marks]

Question 2

(a) A bitonic sequence is is a sequence of numbers $a_0, a_1, a_2, a_3, a_4, , a_{n-1}$ which monotonically increases in value and then decreases i.e.
$a_0 < a_1 < a_2 < \ldots < a_i > a_{i+1} > \ldots > a_{n-1}$, for some i.
A property of a bitonic sequence is that if we compare and exchange operation on between a_i and $a_{i+n/2}$ then we get two bitonic sequences where the numbers in one sequence are all less than the numbers in another sequence.

Use the above property to sort the sequence 3,5,8,9,7,4,2,1 into increasing order. [4 marks]

(b) Explain how what happens at each stage for 2^k numbers. [4 marks]

(c) Explain what the communications requirements are at each stage if each processor holds only one number. [3 marks]

(d) Explain how you implement this algorithm in parallel for 2^k numbers on 2^k processors and what the practical considerations might be. [5 marks]

(d) Calculate how much work each processor does at each stage in terms of communications and computation and hence calculate the speedup and isoefficiency of the algorithm. [4 marks]

Question 3

(a) Describe the odd-even parallel mergesort algorithm and how it may be used to sort an unsorted list of numbers into a sorted list.

[4 marks]

(b) Show how the algorithm would be used to sort the sequence 3,5,8,9,7,4,2,1 into increasing order. [4 marks]

(c) Explain what the communications requirements are at each stage if each processor holds (i) only one number (ii) 2^m numbers . [2 marks]

(d) Consider a bucket sort algorithm in which each processor only has one bucket. Suppose that sorting n numbers with the aid of m buckets takes $nlog(n/m)$ operations. Explain how the algorithn is modified to use p small buckets per processor. [4 marks]

(e) Derive the cost of redistributing p buckets from one processor to all the others and hence calculate the overall cost and then the efficiency. [4 marks]

(f) Which MPI function would you use to implement the redistribution? Where do you think the algorithm may not be scalable on a real calculation? [2 marks]

Question 4

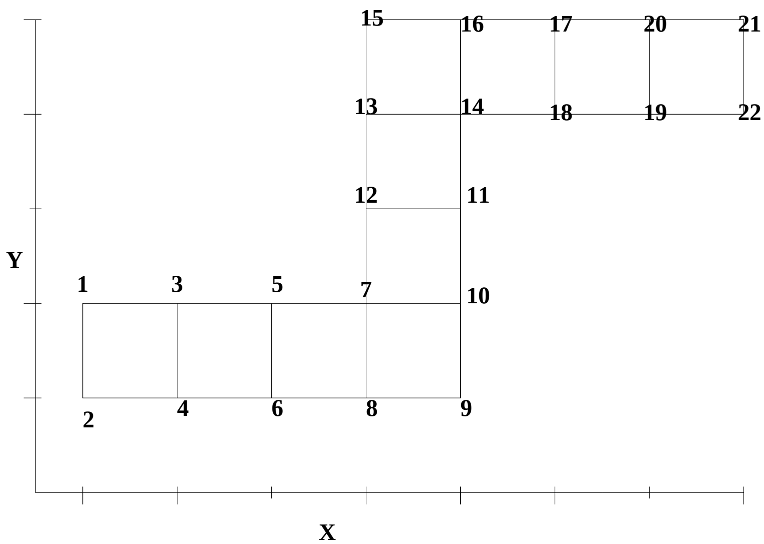
(a) Explain in detail the differences between threads and Unix processes. [2 marks]

(b) Define what is meant by a critical section, a mutex and condition variables. [3 marks]

(c) Describe in outline the main features of an old multithreaded parallel machine (e.g. Tera MTA) and contrast them with a standard shared memory machine. [2 marks]

(d) Explain the importance of graph partitioning techniques for the load balancing of parallel problems. Describe the partitioning methods based on Recursive Co-ordinate Bisection and Space Filling Curves and indicate their advantages and drawbacks. [6 marks]

(e) Apply these two methods to the graph below and in each case produce partitions of the graph



for 2 and 4 processors. (You may use the worksheet at the end of the exam paper) [4 marks]

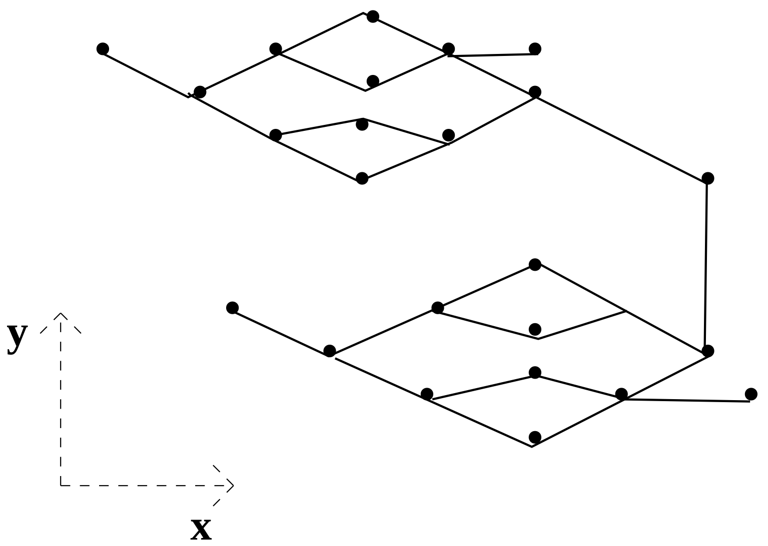
(f) Calculate the cutweights for the 4-way partitions. Do the cutweights match the actual communications volume? [2 marks]

(g) Describe the factors that influence the communications performance and discuss whether or not the graph model matches these factors. [1 mark]

Question 5

(a) Explain the importance of graph partitioning techniques for the load balancing of parallel problems. Describe the partitioning methods based on Recursive Co-ordinate Bisection and Space Filling Curves and indicate their drawbacks. [4 marks]

(b) Apply these two methods to the graph below and in each case produce partitions of the graph



for 2 and 4 processors. [4 marks]

(c) Calculate the cutweights for the 4-way partitions. Do the cutweights match the actual communications volume? [2 marks]

(d) Describe the factors that influence the communications performance and discuss whether or not the graph model matches these factors. [2 marks]

(e) Determine and explain how the following code works and what it is for

```
void unknown()
{
    lock(arrival);
    count++;
    if(count < n)unlock(arrival)
    else unlock(departure);
    lock(departure);
    count--
    if(count > 0)unlock(departure)
    else unlock(arrival);
    return;
}
```

[4 marks]

Question 6

(a) Explain, with reference to OpenMP what is meant by: a team of threads, parallel directives, parallel for loops, private ivariables, shared variables, critical satetements, atomic statements and chunksize. [6 marks]
(b) Explain, how OpenMP is being extended to both gpus and the Intel Xeon Phi (MIC) [2 marks]

(c) Explain how you would implement the following loop in openmp. [8 marks]

```
  emax = 0.0;
/* Gauss Seidel sweep of interior nodes  */
  for (i = 2; i< n1; ++i){
     for (j = 2; j< n1; ++j){
       temp = (h[i-1][j]+h[i+1][j]+h[i][j-1]+h[i][j+1])*0.25;
       err  = err + fabs( h[i][j] - temp);
       if( err > emax) emax = err;
       h[i][j]= temp;
       }
    }
```

Discuss at least two different options.

(d) Given the inherently serial nature of the algorithm above, how would you ensure that the parallel code performed both efficiently and with answers close to the serial one? [4 marks]

Question 7

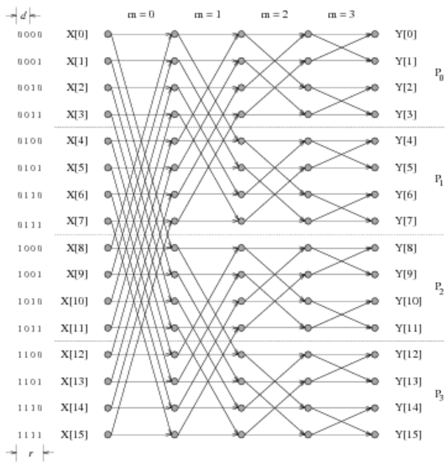


Figure 13.4 A 16-point FFT on four processes. P_i denotes the process labeled i. In general, the number of processes is $p = 2^d$ and the length of the input sequence is $n = 2^r$.

Figure 1: FFT Diagram

(a) Explain, in general terms, how an operation with N (where $N = 2^q$ for some q) elements which may be decomposed into two operations with $N/2$ elements operations may be hierarchically decomposed into $log(N)$ stages. You may use the FFT decomposition as an example. [4 marks]

(b) Explain how you would implement such an operation in parallel, assuming that the decomposition at the jth stage involves each of p processors doing N/p operations and receiving N/p data values from other processors. [4 marks]

(c) Derive the communications and computation costs for using p processors.Include the case $p = 1$ to get a serial result. [4 marks]

(d) Hence calculate the speedup and efficiency of the parallel algorithm [4 marks]

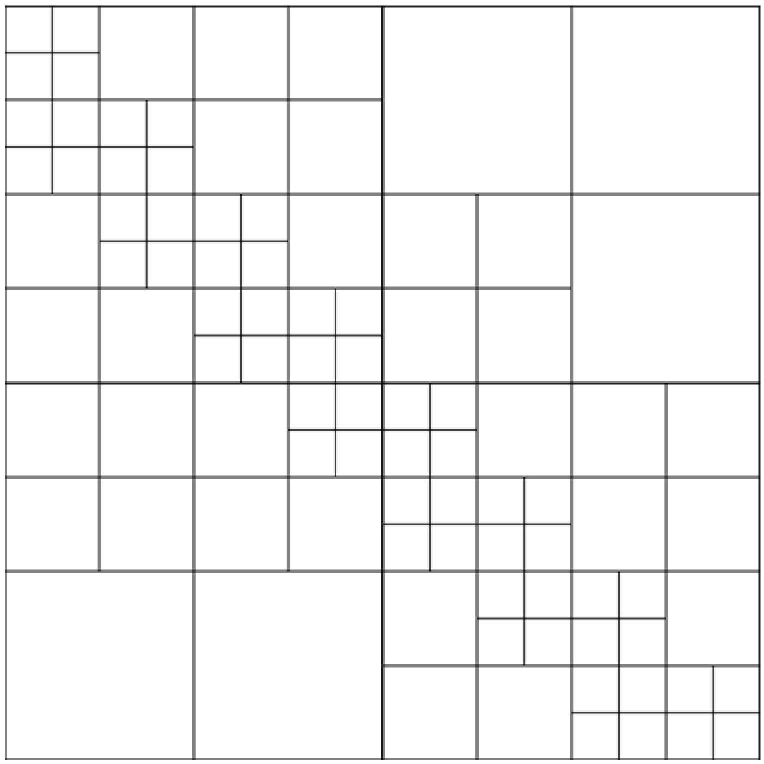
(e) Consider an alternate distribution of initial values in the FFT algorithm in the diagram for four processors as given below.

```
  P0  |   P1  |  P2  |  P3

X(0) |  X(1) | X(2) | X(3)
X(4) |  X(5) | X(6) | X(7)
X(8) |  X(9) | X(10)| X(11)
X(12)|  X(13)| X(14)| X(15)
```

By using the FFT communications paths given in the FFT diagram on the next page show that this decomposition allows the first two parts of the FFT to be completed without interprocessor communication. Then show how the variables should be moved between processors so that the final two parts of the original algorithm may be done without interprocessor communication. Compare this algorithm against the original algorithm. [4 marks]

Question 8



(a) Explain the importance of load balancing parallel codes for solving parallel problems. Describe the partitioning methods based on Recursive Co-ordinate Bisection and Space Filling Curves and indicate their drawbacks. Contrast their approaches with the use of an an explicit graph-based approach [4 marks]

(b) Apply these two methods to the graph above. Assume that we work is notionally located at the centroid of each cell and that the communication takes places across cell edges. In each case produce partitions of the graph for 2 and 4 processors by using the two work sheets attached to this question paper. [4 marks]

(c) Calculate the cut-weights for the 4-way partitions. Do the cutweights match the actual communications volume? [2 marks]

(d) Describe the multilevel graph-based approach used in some of todays load balancers. In particular explain how mesh coarsening is done and how the quality of the partition is improved using the Fiduccia-Mattheyses Algorithm. What is the computational complexity of these algorithms.

[10 marks]

Work Sheet

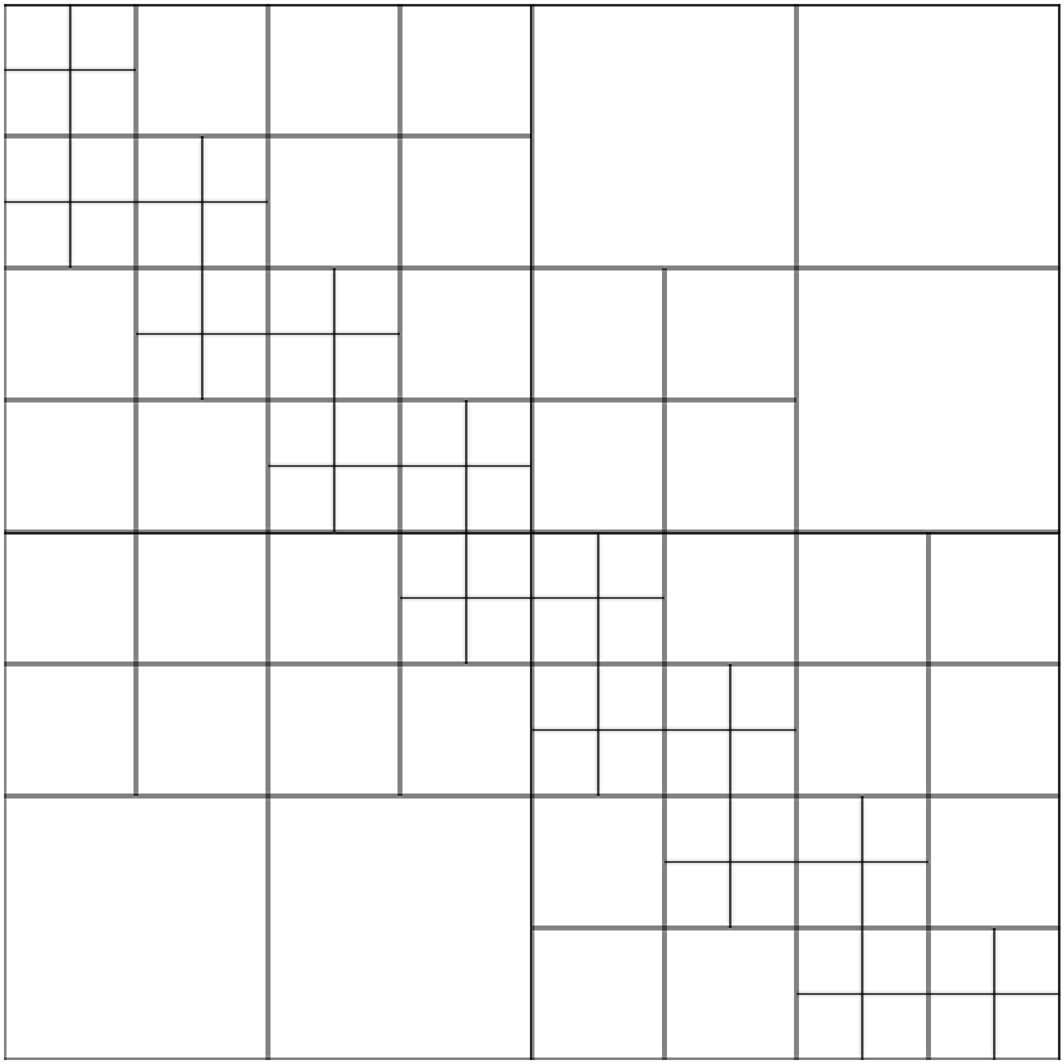


Figure 2: Example using RCB

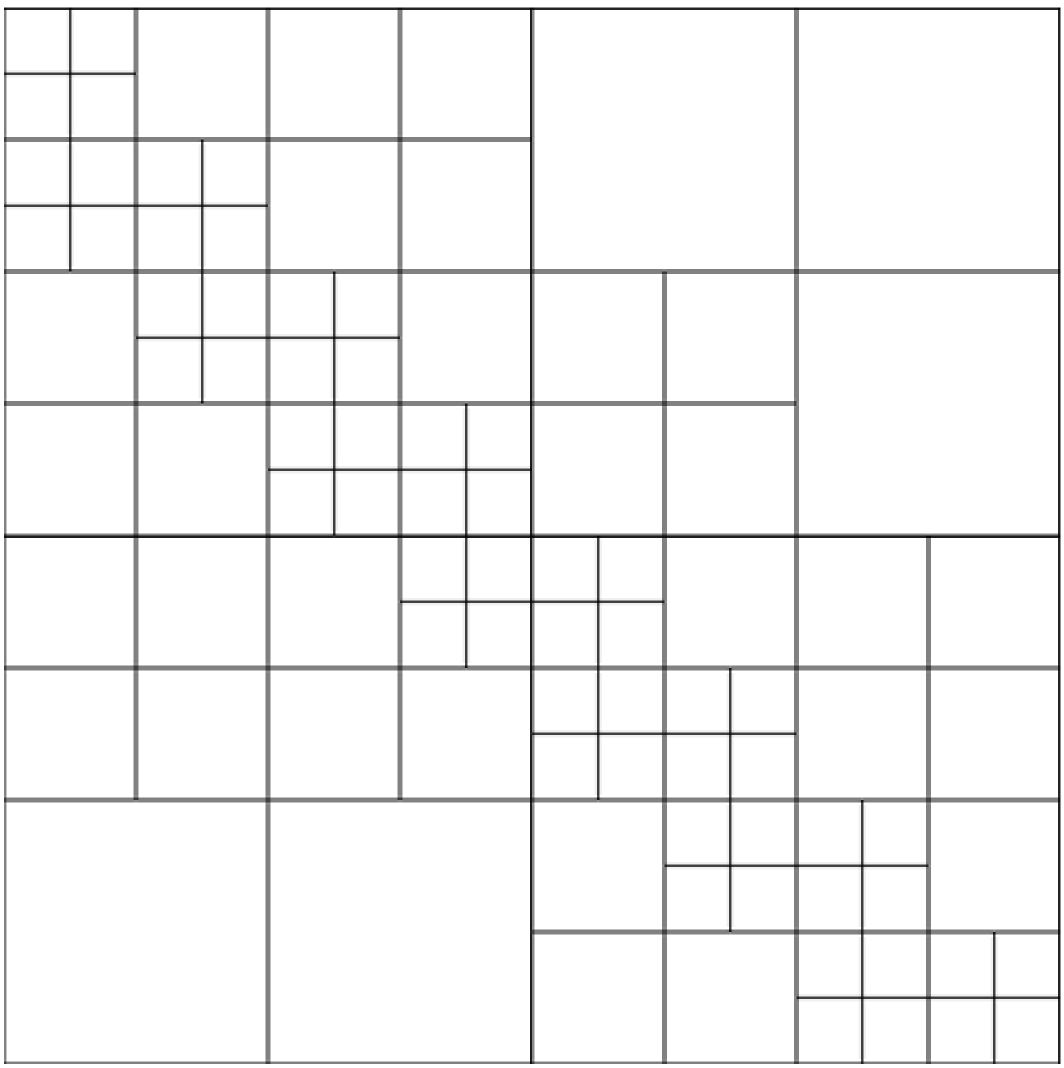


Figure 3: Example using Space Filling Curves