Exam Parallel Programming 1 April 2008
Department of Computer Science, Faculty of Sciences

1. What is Amdahl's law? Give Amdahl's formula and explain what it means and what its implications are.

2. Given below is the pseudo-code for a parallel algorithm that tries to solve the All-pairs Shortest Paths (ASP) problem, i.e. it computes the shortest route between any two cities, where the lengths of the direct routes betweens the cities are given by a NxN matrix C. The send/receive primitives used are FIFO-ordered, so messages between two nodes always arrive in the order they were sent.

   Explain why this algorithm is incorrect and may sometimes compute incorrect results, depending on the timing behavior of the program.

```
int lb, ub;        /* lower/upper bound for this CPU */
int rowK[N], C[lb:ub, N];     /* pivot row ; matrix */

for (k = 1; k <= N; k++) {
   if (k >= lb && k <= ub) {          /* do I have it? */
      rowK = C[k,*];
      for (p = 1; p <= nproc; p++) /* broadcast row */
         if (p != myprocid) SEND(p, rowK);
   } else
      RECEIVE_FROM_ANY(&rowK);        /* receive row */
   for (i = lb; i <= ub; i++)      /* update my rows */
      for (j = 1; j <= N; j++)
         C[i,j] = MIN(C[i,j], C[i,k] + rowK[j]);
}
```

3. What are the advantages and disadvantages of the following interconnection topologies: a 2D mesh, a binary tree, and a hypercube.

4. What is a processor array (vector machine)? Explain how it differs from a multiprocessor and why it is programmed differently.

5. What is the purpose of a 'select statement' (in languages like SR)? Give an example of how this statement can be used.

6. The parallel programming language SR supports (1) implicit and explicit message receipt, and (2) synchronous and asynchronous message sending. Explain which four communication primitives are obtained by combining the different ways of receiving and sending messages in all (four) possible ways. Illustrate the usefulness of each primitive using an example that shows how the primitive can be used.

7. Explain what the DISTRIBUTE directive is used for in HPF (High Performance Fortran). Explain why this directive determines both the amount of communication and the load balancing of HPF programs.

8. The parallel Barnes-Hut algorithm for hierarchical N-body problems can use several different load balancing mechanisms. One approach is to partition the physical space into equal parts and assign all bodies in this part of the space to one processor (so, different processors work on different parts of the physical space). Explain how good or bad both the load balancing and the communication overhead of this approach are. For each of them, explain whether they are better or worse than for the costzone approach.

9. Explain how the load balancing mechanism of TDS (Transposition-Driven Search) works and how it differs from the "work stealing" load balancing mechanism of traditional parallel search methods (like IDA*).

10. Explain why many parallel applications (like N-body simulations and divide-and-conquer algorithms) can obtain good performance on Grids (collections of clusters connected by wide-area networks), despite the fact that the wide-area connections are orders of magnitude slower than local interconnects within clusters (like Myrinet).

Points

1	2	3	4	5	6	7	8	9	10
8	10	10	8	8	10	8	10	8	10

Total: 90 (+ 10 = 100)