CS546 "Parallel and Distributed Processing"
Homework 2

Submission:
Due by 11:59pm of 10/1/2019
Total points 100 - Late penalty: 10% penalty for each day late
Please upload your assignment on Blackboard with the following name:
CS546_SectionNumber_LastName_FirstName_HW2.
Please do NOT email your assignment to the instructor and/or TA!

1. (10 **points**) Suppose a shared-memory system has two cores, core 0 and core 1, both have the variable x in their cache. After core 0 executes the assignment x = 5, core 1 tries to execute y = x. What value will be assigned to y? Why? Can you provide a solution to fix the problem?
2. **(10 points)** Consider a memory system with a level 1 cache of 32 KB and DRAM of 512 MB with the processor operating at 1 GHz. The latency to L1 cache is one cycle and the latency to DRAM is 100 cycles. In each memory cycle, the processor fetches four words (cache line size is four words). What is the peak achievable performance of a dot product of two vectors? Note: Where necessary, assume an optimal cache placement policy.

```
/* dot product loop */
for (i = 0; i < dim; i++)
  dot_prod += a[i] * b[i];
```

3. **(10 points)** Now consider the problem of multiplying a dense matrix with a vector using a two-loop dot-product formulation. The matrix is of dimension 4K x 4K. (Each row of the matrix takes 16 KB of storage.) What is the peak achievable performance of this technique using a two-loop dot-product based matrix-vector product?

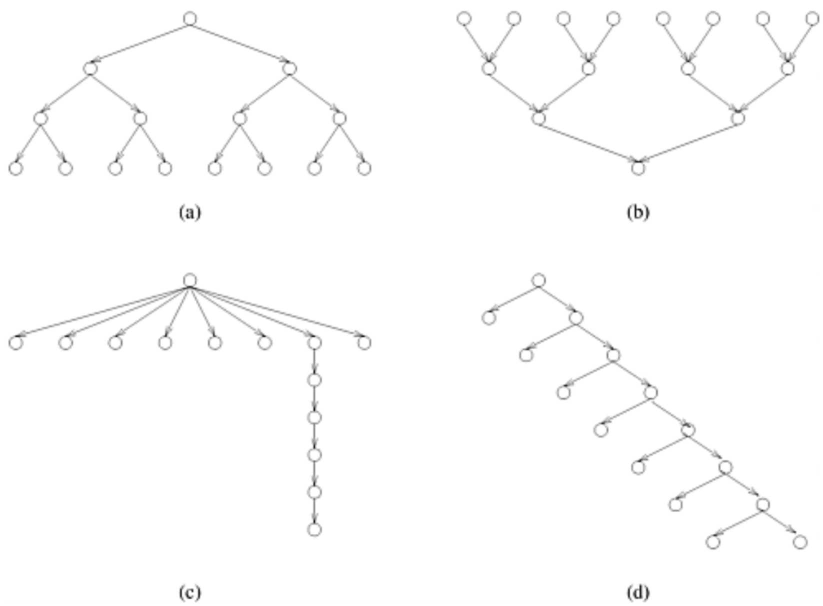
```
/* matrix-vector product loop */
for (i = 0; i < dim; i++)
  for (j = 0; i < dim; j++)
    c[i] += a[i][j] * b[j];
```

4. **(10 points)** Enumerate the critical paths in the decomposition of LU factorization shown in the following figure (textbook figure 3.27).

$$\begin{pmatrix} A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,1} & A_{3,2} & A_{3,3} \end{pmatrix} \rightarrow \begin{pmatrix} L_{1,1} & 0 & 0 \\ L_{2,1} & L_{2,2} & 0 \\ L_{3,1} & L_{3,2} & L_{3,3} \end{pmatrix} \cdot \begin{pmatrix} U_{1,1} & U_{1,2} & U_{1,3} \\ 0 & U_{2,2} & U_{2,3} \\ 0 & 0 & U_{3,3} \end{pmatrix}$$

1: $A_{1,1} \rightarrow L_{1,1}U_{1,1}$	6: $A_{2,2} = A_{2,2} - L_{2,1}U_{1,2}$	11: $L_{3,2} = A_{3,2}U_{2,2}^{-1}$
2: $L_{2,1} = A_{2,1}U_{1,1}^{-1}$	7: $A_{3,2} = A_{3,2} - L_{3,1}U_{1,2}$	12: $U_{2,3} = L_{2,2}^{-1}A_{2,3}$
3: $L_{3,1} = A_{3,1}U_{1,1}^{-1}$	8: $A_{2,3} = A_{2,3} - L_{2,1}U_{1,3}$	13: $A_{3,3} = A_{3,3} - L_{3,2}U_{2,3}$
4: $U_{1,2} = L_{1,1}^{-1}A_{1,2}$	9: $A_{3,3} = A_{3,3} - L_{3,1}U_{1,3}$	14: $A_{3,3} \rightarrow L_{3,3}U_{3,3}$
5: $U_{1,3} = L_{1,1}^{-1}A_{1,3}$	10: $A_{2,2} \rightarrow L_{2,2}U_{2,2}$	

5. **(10 points)** Show an efficient mapping of the task-dependency graph of the decomposition shown in the above figure (textbook figure 3.27) onto three processes, four processes. Prove informally that your mapping is the best possible mapping for three processes.
6. **(20 points)** For the task graphs given in the following figure, determine the following:
   a) Maximum degree of concurrency.
   b) Critical path length.
   c) Maximum achievable speedup over one process assuming, that an arbitrarily large number of processes is available.
   d) The minimum number of processes needed to obtain the maximum possible speedup.
   e) The maximum achievable speedup if the number of processes is limited to 2, 4, and 8.



7. **(20 points)** We have introduced the Parallel Partition LU Tridiagonal (PPT) algorithm in class (see slide 16 of Lecture 8). An algorithm count and communication count analysis result of the PPT algorithm are also presented in Lecture 8. Please provide a detailed, step-by-step algorithm count and communication count analysis of the PPT algorithm to reach the result or reach your own analysis result.
8. **(10 points)** We have introduced the Parallel Diagonal Dominant (PDD) algorithm in class (see slide 29 of Lecture 8). An algorithm count and communication count analysis result of the PDD algorithm are also presented in Lecture 8. Please provide a detailed, step-by-step algorithm count and communication count analysis of the PDD algorithm to reach the result or reach your own analysis result.

Note: We encourage collaboration between you and your classmates. Discuss various approaches and techniques to better understand the questions. However, we do NOT allow copying solutions or code. This is considered as cheating and falls under IIT code of honor. Penalties will be enforced. Please make sure you write your own solutions.
GOOD LUCK!