CS 377P Spring 2017
Assignment 5: Review
Due: April 6th, 2017

March 29, 2017

This assignment should be done individually.

1. *Dependence analysis (25 points)* Draw the dependence graph for the following program. You should specify the type of each dependence in your graph. What are the strongly connected components in this graph?

```
S1 :  A = B+C;
      for I = 1, N {
S2 :      D(I) = A*E(I);
S3 :      S = E(I)*5;
S4 :      T = T+S; //note that the last symbol is "S" and not five
S5 :  A = D(N)-7;
```

2. *Caches (25 points)* This problem is from the book by Hennessy and Patterson.

   (a) (10 points) You are given a direct-mapped cache with 16 one-word blocks. The cache is initially empty. The processor issues the following memory requests in sequence: 1,4,8,5,20,17,19,56,9,11,4,43,5,6,9,17. Label each memory request in this sequence as a hit or a miss, and show the final contents of the cache.

   (b) (15 points) Repeat this exercise for a direct-mapped cache with four-word blocks, and a *total* size of 16 words.

3. *Strongly connected components (25 points)* In a directed graph $G = (V, E)$, nodes u and v are in the same strongly connected component if there is a path from u to v, and a path from v to u.

   (a) (10 points) What are the strongly connected components in the graph of Figure 1?

   (b) (10 points) What is the acyclic condensate of this graph?

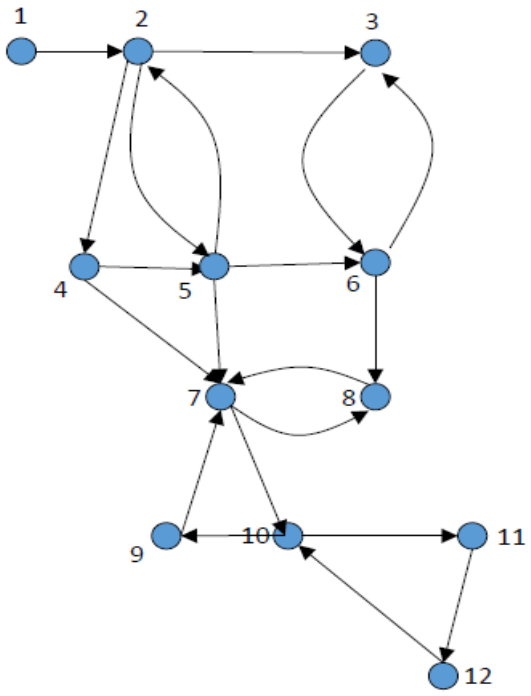

Figure 1: Directed graph for Problem 2

(c) (5 points) What is the asymptotic complexity of the fastest known algorithm for computing strongly connected components?

4. *Caches (25 points)*

(a) (3 points) What is a cold miss? What can you do in hardware to reduce the number of cold misses? What can you do in software?

(b) (3 points) Repeat part (a) for capacity misses.

(c) (3 points) Repeat part (a) for conflict misses.

(d) (5 points) Repeat part (a) for invalidation misses.

(e) (3 points) Explain briefly what "LRU replacement policy" means in the context of caches.

(f) (4 points) Johnny Cache, a CS377P student, says that the concept of "replacement policy" is irrelevant for direct-mapped caches. Is he right? How about for set-associative caches in which the set-size is more than one?

(g) (4 points) Explain briefly why the size of a set in a set-associative cache does not have to be a power of 2.