Practice Problems

1. (a) $T(n) = 3T(n/2) + n$, $T(1) = 1$.

   (b) $T(n) = T(n/2)$. $T(1) = 1$.

2. Suppose you are given an array of n integers. Your goal is to find the largest and the smallest elements of the array. Design an algorithm that reduces the number of comparisons made among the elements in the array. Derive the number of comparisons. Your grade depends on the number of comparisons. It is possible to design an algorithm that does approximately $3n/2$ comparions.

3. Given a directed graph $G = (V, E)$, a vertex $s \in V$ is called *source vertex* if the in-degree of s is zero. Consider the following undirected graph (which is a DAG). Identify all source nodes in the graph. Write a topological order of vertices in the graph. **A DAG of your choice**

4. Prove that every DAG has at least one source vertex.

5. Consider the following code fragment:

   ```
   int i = 1;
   int s = 1;
   while (i <= n) {
     s = s+ 2i+1;
     i = i+1;
   }
   ```

   Show that at the start of every iteration $s = i^2$ holds using induction.

6. You are in a rectangular maze organized in the form of $M \times N$ cells/locations. You are starting at the upper left corner (grid location: $(1, 1)$) and you want to go to the lower right corner (grid location: (M, N)). From any location, you can move either to the right or to the bottom, or go diagonal. I.e., from (i, j) you can move to $(i, j + 1)$ or $(i + 1, j)$ or to $(i + 1, j + 1)$. Cost of moving right or down is 2, while the cost of moving diagonally is 3. The grid has several cells that contain diamonds of whose value lies between 1 and 10. I.e, if you land in such cells you earn an amount that is equal to the value of the diamond in the cell. Your objective is to go from the start corner to the destination corner by picking up diamonds such that the difference between the sum of the value of diamonds you picked and sum of the costs incurred is maximized. Write a dynamic programming algorithm, based on appropriate recurrence relation. State the recurrence, algorithm and analyze its time complexity.