# Time: 70 minutes. Closed books. Closed notes.

**You are required to shut down and put all electronic devices away. Make sure that cell phones don’t make any noise during the exam.**

**Your algorithms should be as efficient as possible. Your algorithms and answers, in general, should also be as simple as possible.**

**Explain your answers.**

**All the questions have equal weight.**

# Good Luck!

 Your Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Honor Pledge**

The university has a nationally recognized Honor Pledge, administered by the Student Honor Council. The Student Honor Council proposed and the university Senate approved an Honor Pledge. The University of Maryland Honor Pledge reads:

"I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination."

 Please write the exact wording of the Pledge followed by your signature in the space below:

Pledge: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Your signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Problem 1

You are asked to radix-sort n integers whose range is 0 to k-1, k > n. How many rounds of count sort do you need? You answer should be a simple formula in terms of k and n.

# Problem 2

G=(V,E) is a connected undirected graph. Each edge e in E has edge weight w(e). Suppose that T, a minimum spanning tree (MST) of G, is also given.

Now, suppose that the weight of a single edge echanges from w(e) to some other value w’(e).

(a) For each of the following two scenarios, describe an (efficient) algorithm to transform the original MST T into T’, an MST that reflects the edge weight change.

(i) e $\in $ T

1. e $\notin $ T

(b) Why is your algorithm correct?

(c) What is the time complexity of your algorithm (as a function of n and m, the number of vertices and edges, respectively)?

# Problem 3

Assume a curriculum consisting of n semester-long courses, all of them mandatory. In the pre-requisite graph G(V, E), the set of vertices V has a vertex for each course, and E has a directed edge from course v to course w if and only if v is a prerequisite for w.

1. Give an (efficient) algorithm that computes the minimum number of semesters necessary to complete the curriculum, assuming that a student can take any number of courses in one semester.
2. Why is your algorithm correct?
3. What is the time complexity of your algorithm (as a function of n and m, the number of vertices and edges, respectively)?

**Problem 4**

An **independent set** of a graph G=(V,E) is a subset V’ of V such that if the edge (u,v) is in E then V’ can include vertex u, or vertex v, or none, but not both.

1. Define a decision problem version of the independent set problem.

Below, you can assume that this independent set problem is NP-Complete, but you are not allowed to assume that any other problem is NP-Complete.

1. Prove that the clique problem is NP-Complete.

#

# Problem 5

You are given a connected undirected graph with weighted vertices. Imagine the weights represent altitude and you need to figure a route for biking from some vertex u to another vertex v. Biking uphill is hard, but going downhill or without change of altitude is easy.

1. Give an (efficient) algorithm that determines which path from u to v requires the least amount of total uphill biking, in terms of *total altitude gained*. (For example, consider a path from u1 to u4 that goes through u2 and u3. If w(u1)=1, w(u2)=8, w(u3)=5 and w(u4)=9, then the total altitude gained is (8-1) + (9-5) = 11.)
2. Why is your algorithm correct?
3. What is the time complexity of your algorithm (as a function of n and m, the number of vertices and edges, respectively)?

# Problem 6

Describe two methods of resolving collisions in a hash table. Give one advantage that each has over the other.