CS 373: Combinatorial Algorithms, Spring 1999

Midterm 2 (April 6, 1999)

Name:	
Net ID:	Alias:

This is a closed-book, closed-notes exam!

If you brought anything with you besides writing instruments and your $8\frac{1}{2}'' \times 11''$ cheat sheet, please leave it at the front of the classroom.

- **Don't panic!**
- Print your name, netid, and alias in the boxes above, and print your name at the top of every page.
- **Answer four of the five questions on the exam.** Each question is worth 10 points. If you answer more than four questions, the one with the lowest score will be ignored. **1-unit graduate students must answer question #5.**
- Please write your answers on the front of the exam pages. You can use the backs of the pages as scratch paper. Let us know if you need more paper.
- Read the entire exam before writing anything. Make sure you understand what the questions are asking. If you give a beautiful answer to the wrong question, you'll get no credit. If any question is unclear, please ask one of us for clarification.
- Don't spend too much time on any single problem. If you get stuck, move on to something else and come back later.
- Write *something* down for every problem. Don't panic and erase large chunks of work. Even if you think it's nonsense, it might be worth partial credit.
- **Don't panic!**

#	Score	Grader
1		
2		
3		
4		
5		

1. **Bipartite Graphs**

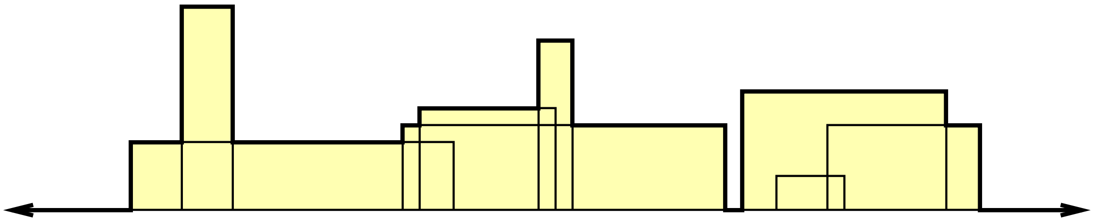
   A graph (V, E) is *bipartite* if the vertices V can be partitioned into two subsets L and R, such that every edge has one vertex in L and the other in R.

   (a) Prove that every tree is a bipartite graph.

   (b) Describe and analyze an efficient algorithm that determines whether a given connected, undirected graph is bipartite.

2. **Manhattan Skyline**

   The purpose of the following problem is to compute the outline of a projection of rectangular buildings. You are given the height, width, and left x-coordinate of n rectangles. The bottom of each rectangle is on the x-axis. Describe and analyze an efficient algorithm to compute the vertices of the "skyline".



A set of rectangles and its skyline.

3. **Least Cost Vertex Weighted Path**

   Suppose you want to drive from Champaign to Los Angeles via a network of roads connecting cities. You don't care how long it takes, how many cities you visit, or how much gas you use. All you care about is how much money you spend on food. Each city has a possibly different, but fixed, value for food.

   More formally, you are given a directed graph $G = (V, E)$ with nonnegative weights on the *vertices* $w \colon V \to \mathbb{R}^+$, a source vertex $s \in V$, and a target vertex $t \in V$. Describe and analyze an efficient algorithm to find a minimum-weight path from s to t. [Hint: Modify the graph.]

4. **Union-Find with Alternate Rule**

   In the UNION-FIND data structure described in CLR and in class, each set is represented by a rooted tree. The UNION algorithm, given two sets, decides which set is to be the parent of the other by comparing their ranks, where the rank of a set is an upper bound on the height of its tree.

   Instead of rank, we propose using the *weight* of the set, which is just the number of nodes in the set. Here's the modified UNION algorithm:

```
UNION(A, B):
    if weight(A) > weight(B)
        parent(B) ← A
        weight(A) ← weight(A) + weight(B)
    else
        parent(A) ← B
        weight(B) ← weight(A) + weight(B)
```

   Prove that if we use this method, then after any sequence of n MAKESETs, UNIONs, and FINDs (with path compression), the *height* of the tree representing any set is $O(\log n)$.

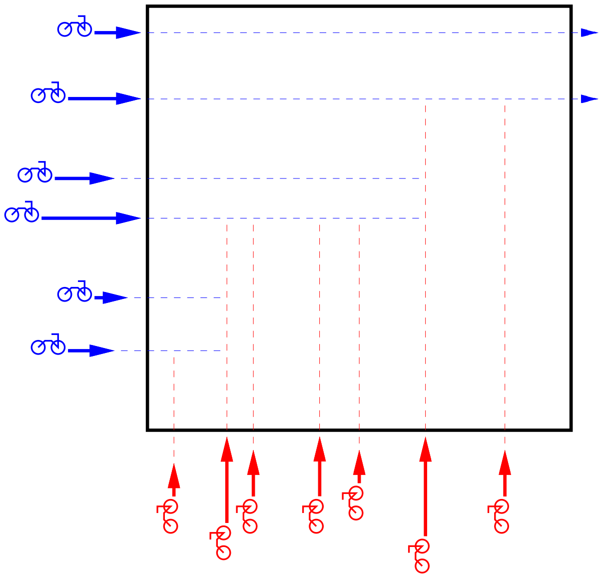
   [Hint: First prove it without path compression, and then argue that path compression doesn't matter (for this problem).]

5. **Motorcycle Collision**

One gang, Hell's Ordinates, start west of the arena facing directly east; the other, The Vicious Abscissas of Death, start south of the arena facing due north. All the motorcycles start moving simultaneously at a prearranged signal. Each motorcycle moves at a fixed speed—no speeding up, slowing down, or turning is allowed. Each motorcycle leaves an oil trail behind it. If another motorcycle crosses that trail, it falls over and stops leaving a trail.

More formally, we are given two sets H and V, each containing n motorcycles. Each motorcycle is represented by three numbers (s, x, y): its speed and the x- and y-coordinates of its initial location. Bikes in H move horizontally; bikes in V move vertically.

Assume that the bikes are infinitely small points, that the bike trails are infinitely thin lie segments, that a bike crashes stops *exactly* when it hits a oil trail, and that no two bikes collide with each other.



Two sets of motorcycles and the oil trails they leave behind.

(a) Solve the case $n = 1$. Given only two motorcycles moving perpendicular to each other, determine which one of them falls over and where in $O(1)$ time.

(b) Describe an efficient algorithm to find the set of all points where motorcycles fall over.

5. **Motorcycle Collision (continued)**

   Incidentally, the movie *Tron* is being shown during Roger Ebert's Forgotten Film Festival at the Virginia Theater in Champaign on April 25. Get your tickets now!