

COMP 482, Fall 2008, Exam 2

You have **5 contiguous hours** of your choice to look at and work on the exam. Hand it in during class or at my office, DH 3118. Remember that evening and weekend access to my office is limited.

You may use any material provided as part of this course, your textbook, notes, and graded assignments, but no other materials. You may not use a computer, except to access the course's online material or to type your answers. Please write legibly, clearly label each problem, and show your work.

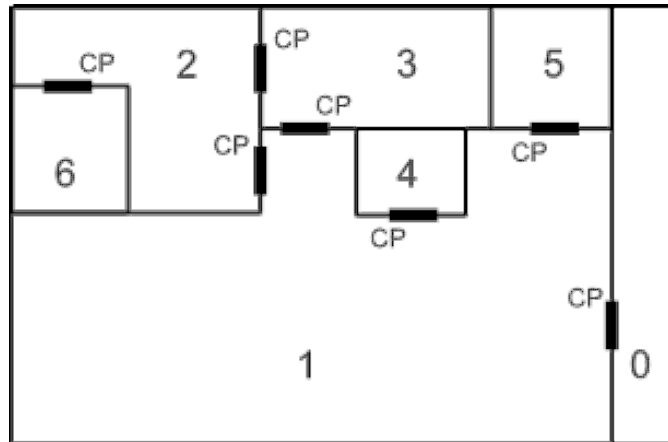
Since this is a take-home exam, it is not practical to ask for clarifications during the exam. If you feel that a problem is not sufficiently specified, clearly state the assumptions that you are making, and continue with the exam. Of course, your assumptions should be reasonable ones.

The 4 problems follow on separate pages.

1. (20 points)

You are creating the Securisys 9043, the latest home security software from Spishak Inc. (Their motto is “So secure, even **you** can’t unlock it.”) The software is designed to “secure” a room; it does this by determining the minimum number of locks it has to perform to prevent access to a given room from one or more other rooms. Each door connects a pair of rooms and has a single control panel that can unlock it from one side of the door.

For example, in the following diagram, each room is numbered, and each control panel (“CP”) is marked next to the door it can unlock and in the room that it is accessible from.



To secure room 2 from an intruder in room 1, the minimum number of locks needed is two: one has to lock the door between room 2 and room 1 and the door between room 3 and room 1. Note that it is impossible to secure room 2 from an intruder in room 3, since the intruder would be able to use the control panel in room 3 that unlocks the door between room 3 and room 2.

Describe how to determine the minimum number of locks needed, given a building layout, the room you are trying to secure, and the locations of one or more intruders.

2. (20 points)

A total of n friends want to go to a free concert, which is d miles away. They are poor college students without cars and only one single-seat bicycle among them. They are in a hurry and want to get to there as soon as possible for the best seats. Since they want to sit together, they are willing to wait until the last friend gets there.

Each person has their own walking speed w_i and biking speed b_i , each in miles per hour. While the friends start at the same time, in order to get there as quickly as possible, they are willing to split up en route.

The following is an example schedule when $n = 3$, $d = 10$, $w_1 = w_2 = 2$, $w_3 = 4$, $b_1 = b_2 = 12$, and $b_3 = 16$.

- Person 1 bikes from mile 0 to mile 5.4 in $\frac{9}{20}$ hours, then walks the rest of the way to mile 10.

- Person 3 walks from mile 0 to mile 5.4 in $\frac{27}{20}$ hours, gets the bike that's now lying there, bikes *back* to mile 4.6 in $\frac{1}{20}$ hour, and then walks to mile 10.
- Person 2 walks from mile 0 to mile 4.6 in $\frac{23}{10}$ hours, gets the bike that's now lying there, and bikes to mile 10.

The bike is so cheap and ugly that no one ever wants to steal it, so it doesn't move unless one of the friends is riding it. In this schedule, all three people arrive simultaneously in 11/4 hours. That this schedule is optimal is not at all obvious.

Create and explain a linear program that determines the minimum time needed for the last person to arrive. Determining the optimum schedule is harder, and not requested. Your LP doesn't need to be in standard form.

You will use the constants n , d , w_i , and b_i , described above. You will also use the following variables:

- t is the total time needed.
- v_i is the time spent walking towards the goal.
- x_i is the time spent walking away from the goal.
- y_i is the time spent biking towards the goal.
- z_i is the time spent biking away from the goal.

In each occurrence, i ranges over the set of people, $\{1, \dots, n\}$.

3. (30 points)

You are given a set P of points in the plane and a radius r , and you are to find the smallest number of disks of radius r that cover all the points.

Prove that this problem is NP-complete. (Hint: Think about its relation to the set cover problem.)

4. (30 points)

Prove that you can color a 3-colorable graph with $O(\sqrt{n})$ colors in polynomial time. (Extended hint: First show that in a 3-colorable graph, the neighbors of any given vertex are 2-colorable. Then show that any graph either has maximum degree \sqrt{n} or has a set of \sqrt{n} vertices that you can take care of with only two colors. This should give you a recursive algorithm.)