CS140: Exam 2 Name ______________________________
11/11/99

This exam consists of 4 problems. The space provided should be adequate for your solutions. Each problem is worth 25 points. Your answers will be judged for clarity, conciseness, and correctness. Bounds and algorithms should be, respectively, as tight and as fast as possible. You may consult any published text and your class notes to complete this exam. You may not discuss the exam (at all) with anyone except me. The exam is due at the start of class on Monday, November 14, 1999.

1. Given an unparenthesized expression over the integers with the operations of addition, subtraction, and multiplication, we'd like to find a way to parenthesize the expression to maximize its value.

   (a) (2 points) For example, the formula $2+3\cdot 6-5$ can be parenthesized many ways including $(((2+3)\cdot 6)-5)=25$, $((2+(3\cdot 6))-5)=15$ and $((2+3)\cdot(6-5))=5$. How should the expression be parenthesized to maximize its evaluation? To minimize its evaluation?

   (b) (6 points) Give a recursive definition for the maximum possible evaluation of the expression $x_1 o_1 x_2 o_2 \ldots x_{n-1} o_{n-1} x_n$, where each x_i is an integer and each o_i is in the set $\{+,-,\times\}$. (Don't forget about the base case(s).)

(c) (2 pts.) Analyze the running time of the recursive algorithm implied by the previous answer.

(d) (12 points) Now we want to construct a dynamic programming algorithm to compute the maximum possible evaluation for an input expression. To describe your algorithm it is sufficient to answer the following questions. (To receive full credit your algorithm should run in time $O(n^3)$.)

i. (2 pts.) Describe the intermediate results you intend to save.

ii. (2 pts.) How do you compute one such result?

iii. (2 pts.) In what order do you compute the results?

iv. (2 pts.) What are the base cases and how do you compute them?

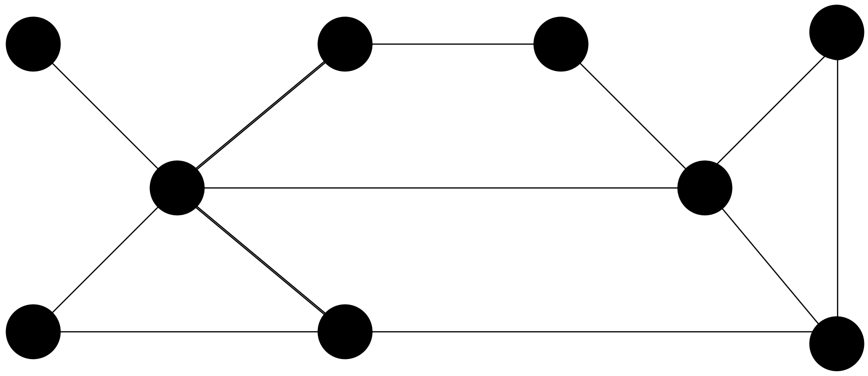
v. (2 pts.) How do you find the answer when you are done?

vi. (2 pts.) How much time does it take to compute one result? How many results do you have to compute? What is the running time of your algorithm?

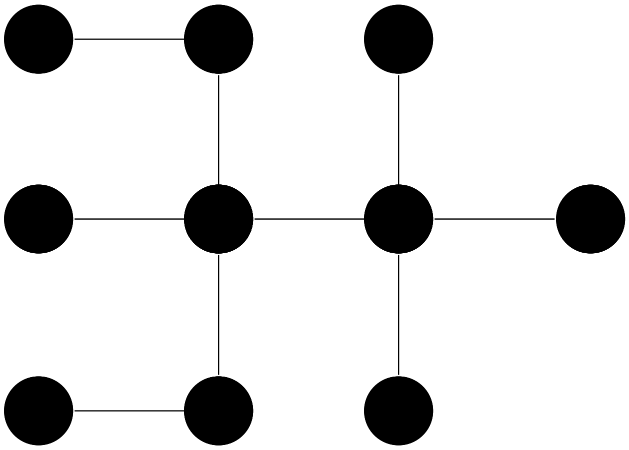
(e) (3 points) Explain how to modify your algorithm so that it outputs a parenthesization of the input expression that yields the maximum evaluation.

2. A dominating set of a graph G is a subset W of the vertices of the graph such that every vertex v of G is either in W or is connected by an edge to a vertex in W.

   (a) (2 points) Find a smallest dominating set of the following graph.

   (b) (2 points) Finding the smallest dominating set of a graph is quite difficult but the problem becomes easier when the input is restricted to be a tree. Find a smallest dominating set in the following tree.

(c) (18 points) Next we want an $O(n)$ dynamic programming algorithm for the problem of finding the size of a smallest dominating set in an input tree. (Hint: Root the tree.)

i. (5 pts.) Describe the intermediate results you intend to save.

ii. (5 pts.) How do you compute one such result?

iii. (2 pts.) In what order do you compute the results?

iv. (2 pts.) What are the base cases and how do you compute them?

v. (2 pts.) How do you find the answer when you are done?

vi. (2 pts.) How much time does it take to compute one result? How many results do you have to compute? What is the running time of your algorithm?

(d) (3 pts.) Now modify your algorithm to construct a smallest dominating set.

3. HMC is trying to put together a team for the international NeedYouAsk competition. The possible team players are x_1, x_2, x_3, x_4. There are, however, several constraints. First of all, x_1 and x_2 are the only ones that talk in their sleep, so at least one of them should be on the team. Unfortunately x_1 would rather translate the big, white beast into Urdu than be on the team with x_2. Also, x_1 and x_3 have extremely knobby heads, with all the advantages that implies, but there isn't enough room in the Mudd-mobile for x_1's sheep and x_3's socks. Finally x_4 can't sort spaghetti unless x_3 is watching, so if x_3 isn't on the team there is no reason for x_4 to be on the team. You can see our problem. We've summarized the constraints as follows:

1. x_1 on the team or x_2 on the team.

2. x_1 off the team or x_2 off the team.

3. x_1 on the team or x_3 on the team.

4. x_1 off the team or x_3 off the team.

5. x_3 on the team or x_4 off the team.

(a) (2 points) Find a a team that satisfies all of the constraints.

(b) (15 points) This problem is going to come up year after year so we'd like you to design an algorithm to solve it. The input will be the number of potential players, n, and the constraints $c_1, \ldots, c_m$. Each c_i involves two potential players x_k and x_j and has one of the possible forms:

x_k on the team or x_j on the team
x_k on the team or x_j off the team
x_k off the team or x_j on the team
x_k off the team or x_j off the team

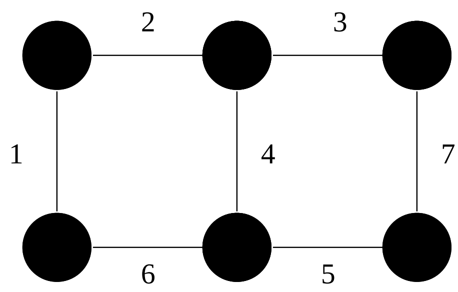
Your algorithm should output a team that satisfies all of the constraints (the empty team is a valid team) or reply that no such team exists. We really hope your algorithm is fast, like $O(n + m)$.

(c) (5 points) Prove your algorithm is correct.

(d) (3 points) Show how to modify your algorithm to determine whether there is a non-empty team that satisfies all of the constraints.

4. Let G be a connected, weighted graph with unique edge weights. For any simple (non-empty) path P in G, $f(P)$ denotes the maximum weight edge in P. For any pair of distinct vertices x and y in G, we define $g(x, y)$ to be the minimum weight edge in the set $\{f(P) : P \text{ is a simple path between } x \text{ and } y\}$.

   (a) Find the edges $g(x, y)$ for each pair of vertices in the following graph.

   (b) Prove that $\cup_{x,y:x \neq y}\, g(x, y)$ is a minimum spanning tree of G.