CS140: Final Exam Name ______________________________

12/10/99

The space provided should be adequate for your solutions. 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 texts and your class notes to complete this exam. You may not discuss the exam (at all) with anyone except me.

1. (5 points) Label the following statements as True or False. Assume that $f(n)$ and $g(n)$ are positive-valued, monotonically-increasing functions on the natural numbers.

   (a) If $f(n) = O(g(n))$ then $\log(f(n)) = O(\log(g(n)))$.

   (b) If $\log(f(n)) = O(\log(g(n))$ then $f(n) = O(g(n))$.

   (c) $f(n) + g(n) = \Omega(\max(f(n), g(n)))$.

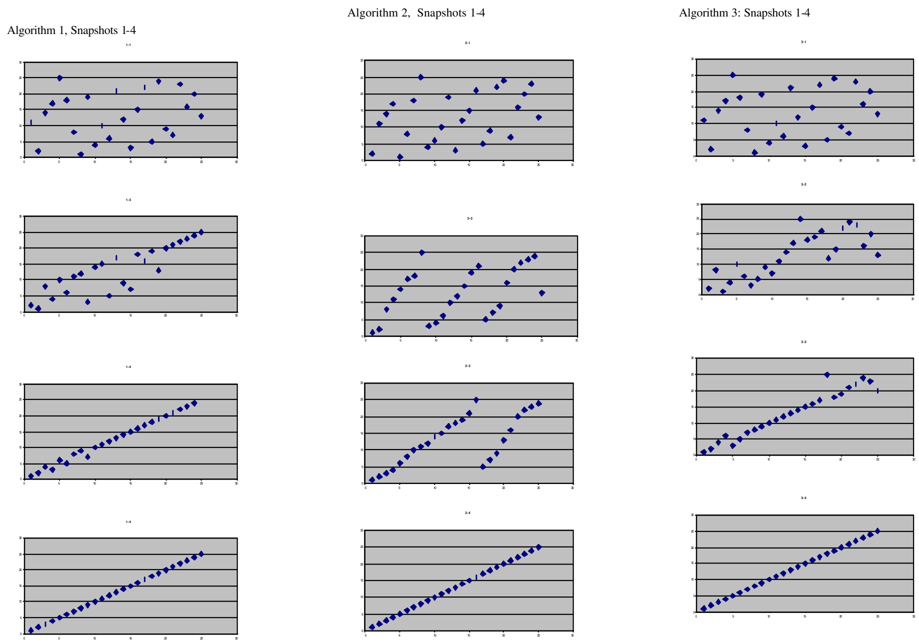
   (d) If $f(n) = O(n^c)$ for some constant c then $f(n) = O(f(n/2))$.

   (e) $f(n) - o(f(n)) = \omega(f(n))$

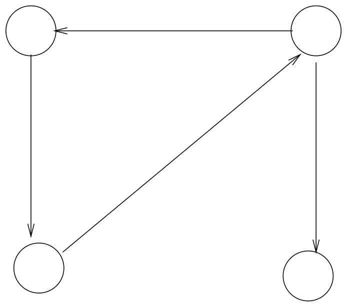
2. (5 points) Order the following functions by increasing rate of growth. You should assume that $0 < \epsilon < 1$ and $c \geq 2$.

$$n^{\epsilon},\ \log(n^c),\ (\log n)^c,\ n^{c/\lg n},\ (\sqrt{c})^{\frac{\lg n}{\epsilon}}$$

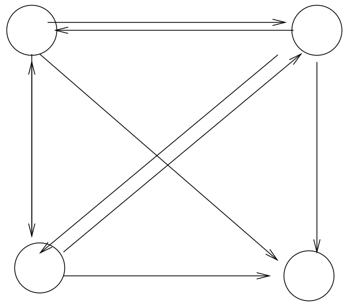
3. (5 points) You are the UBS of company X. The success of X's next generation product line hinges on a fast algorithm for the problem NeedYouAsk. Your programmers come to you with two algorithms. The first has a worst-case running time of $5 \cdot 2^n$ but a guaranteed average-case performance of $20n + o(n)$. The second takes $2n^2 + 3n + o(n)$ steps on all inputs of size n. It is your decision. What algorithm will you use? Give your rationale.



4. (5 points) An array consisting of the integers 1 through 25 in random order was input to three different sorting algorithms. Snapshots of the array during the sorting process are shown above. Each chart shows the value (y-axis) in each array position (x-axis). For each column, name the sorting algorithm that was used.



G



Transitive Closure of G

5. (25 points) Let $G = (V, E)$ be a directed graph. The transitive closure of G, denoted G^T, is the graph with vertex set V and edge set

$$E^T = \{<u, w> \ | \text{ there is a path from } u \text{ to } w \text{ in } G\}.$$

An example is shown in the above figure.

Let $d(u, w)$ denote the length of the shortest path from vertex u to vertex w in G. The diameter of G is defined to be

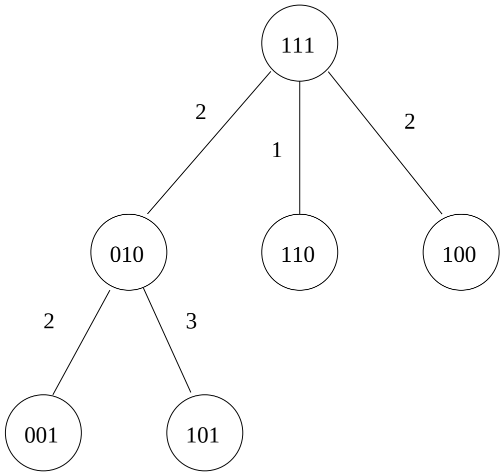
$$max\{d(u, w)| \text{ there is a path from u to v in } G\}.$$

The diameter of the graph G in the above figure is 2.

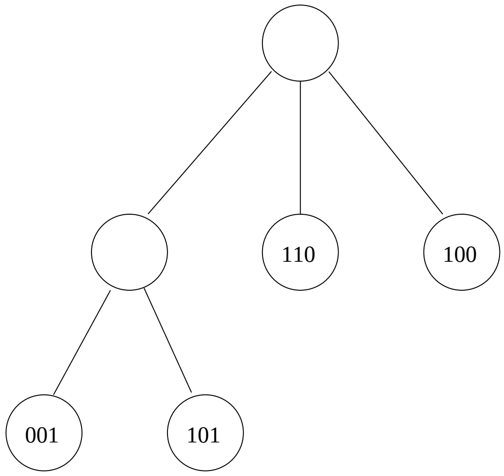
(a) (10 points) Give an algorithm that takes as input the adjacency matrix M and diameter d of a directed graph G and outputs the adjacency matrix of G^T. To receive full credit your algorithm should run in time $O(d \cdot n^{\lg 7})$, where n is the number of vertices of G. (You may assume that n is a power of 2.)

(b) (10 points) Give an algorithm for the case when G is acyclic that runs in time $O(n^{\lg 7})$. (Hint: Show how to renumber of the vertices of G so that M is upper-triangular. Then use recursion and the result from part(a).)

(c) (5 points) Give an algorithm for the general case that runs in time $O(n^{\lg 7})$. (Hint: Give an efficient reduction to the acyclic case.)



Labeled tree



Leaf-labeled tree

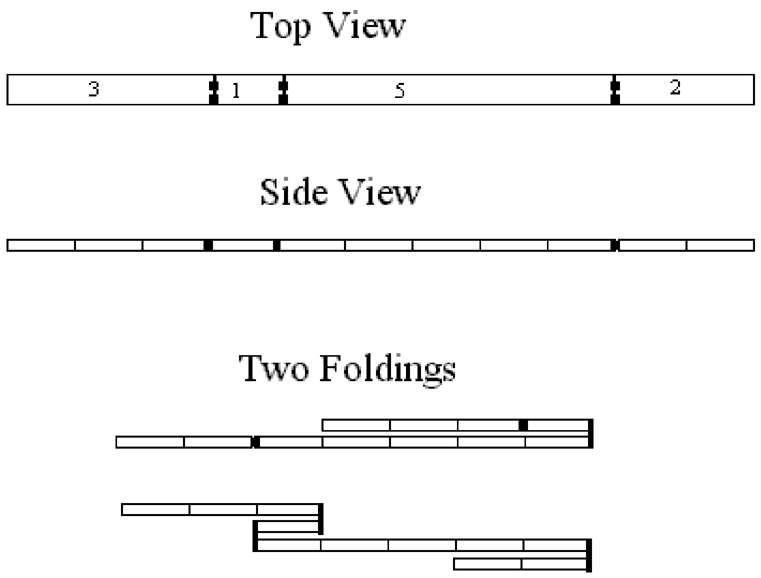
6. (15 points) Let T be a rooted tree in which each node v is labeled by a binary string, $L(v)$, of length k. An example where $k = 3$ is shown above (left). The cost of an edge $e = (u, v)$, denoted $c(e)$, is defined to be the number of positions in which $L(u)$ and $L(v)$ differ. The cost of T is $c(T) = \sum_{e \in T} c(e)$. The edge costs are shown for the labeled tree above. The cost of this tree is 10.

   Given a rooted tree in which each leaf is labeled by binary strings of length k, we'd like to find a way to label the interior nodes so as to minimize the overall cost of the tree.

   (a) (1 points) Label the internal nodes of the tree in the figure above (right) in a way that minimizes cost.

   (b) (7 points) Given an $O(n)$ algorithm for this problem for the case $k = 1$, where n is the number of nodes in T.

(c) (7 points) Generalize your algorithm to arbitrary k. What is the running time of your algorithm?



7. (20 points) A folding ruler is shown in the figure above (top view and side view). Such a ruler consists of a sequence of wooden segments attached end-to-end by hinges. The hinges allow the ruler to be folded in a variety of ways. Two possible foldings are shown above.

   Given such a ruler, we would like to know how compactly it can be folded.

   Folding Ruler Problem (FRP):

   Input: Integer lengths of the segments of a folding ruler: $L_1, L_2, \ldots, L_n$ and an integer K.

   Question: Can the ruler be folded so its total length is no more than K?

   (a) (1 point) Enumerate all possible folded lengths of the ruler shown in the figure above.

(b) (2 points) Give a yes-instance and a no-instance of the FRP.

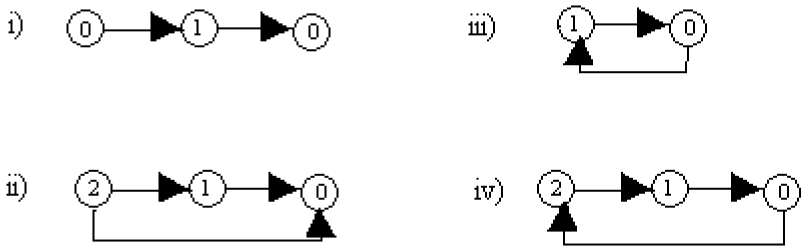
(c) (2 points) Show that FRP is in NP. (Clearly define the witness your verification algorithm expects.)

(d) Show that FRP is NP-complete by giving a polynomial-time reduction from Partition.

   i. (5 points) Let $S = \{x_1, \ldots, x_n\}$ be an instance of Partition. Describe your algorithm for producing the associated instance of FRP. (Be sure to show it is a polynomial-time reduction.)

ii. (5 points) Suppose the instance S is a yes-instance of Partition. Prove that the instance of FRP that is produced by your reduction is also a yes-instance.

iii. (5 points) Suppose the instance of FRP produced by your reduction is a yes-instance. Prove that the original instance of partition is also a yes-instance.



8. (20 points) Let $G = (V, E)$ be a directed graph and let $L : V \longrightarrow \{0, 1, \ldots\}$ be a labeling of the vertices of G. We say that L is stable if for each vertex $u \in V$, $L(u)$ is the smallest non-negative integer that is **not** in the set $\{L(w) \mid <u, w> \in E\}$.

   (a) (2 points) Which of the labelings in the above figure are stable?

   (b) (3 points) Suppose G is a simple directed cycle on n vertices. Prove that G has a stable labeling if and only if n is even.

(c) (5 points) Suppose $G_1 = (V_1, E_1)$ is a simple directed cycle on an odd number of vertices and $G_2 = (V_2, E_2)$ is arbitrary digraph. Let u_1 be a vertex of G_1 and let u_2 be a vertex of G_2. Prove that the digraph

$$G_3 = (V_1 \cup V_2, E_1 \cup E_2 \cup \{< u_1, u_2 >\}$$

has a stable labeling if and only if G_2 has a stable labeling in which $L(u_2) = 0$.

(d) (10 points) Give a polynomial-time reduction from 3SAT to the following problem. Prove your reduction is correct. (This part is fairly easy if you've understood the previous parts.)

Stable Label:

Input: Digraph G

Question: Does G have a stable labeling?