CS140 Final Exam

May 2, 2001

This is a 24 hours take-home exam. If you are a senior, you must submit the exam by 8 A.M., Thurday, May 3. Otherwise you must submit the exam by the end of the day on Friday, May 11. This is a closed book/closed notes exam. You may use a one page cheat sheet to help you. You may not discuss the exam with anyone.

Your answers will be judged for clarity, conciseness, and correctness. Bounds and algorithms should be, respectively, as tight and as fast as possible. You have more than adequate time to write your answers carefully. Illegible, confusing, and/or incomplete ideas will be judged harshly. The space provided should be adequate for your solutions.

Fill in your name and when you started and finished the exam in the spaces provided. Good luck!

Name: ______________________________

Start Day/Time ______________________________

End Day/Time ______________________________

Part A: The first part of the exam consists of 25 short-answer questions. Each is worth 1 point. The space provided should be adequate for your solutions.

1. Rank the following functions by increasing rate of growth:

$$(\lg n)^{\lg n}, \quad n/\lg n, \quad n^{3/2}, \quad \lg n^{\lg n}$$

2. Which of the following functions are $\Theta(lg(n))$?

$$\lg(2n), \quad \lg(n^2), \quad \lg^2(n), \quad \lg(\sqrt{n}), \quad \ln(n)$$

3. Give a tight bound for the following recurrence relation:

$$T(n) = 3T(n/2) + cn^2$$

4. Give an example of a monotonically increasing function on the natural numbers, $f(n)$, such that $f(2n) \neq O(f(n))$.

5. Give an example of a monotonically increasing function on the natural numbers, $f(n)$, such $f(2n) = O(f(n))$.

6. Is the following statement true or false? If for every n, for some input of size n Algorithm A takes $3n^2$ steps then A is $\Omega(n^2)$.

7. Is the following statement true or false? If for every n, for some input of size n Algorithm A takes $3n^2$ steps then A is $O(n^2)$.

8. Is the following statement true or false? If $f(n)$ and $g(n)$ are monotonically increasing functions on the natural numbers then $f(n) = O(g(n))$ or $g(n) = O(f(n))$.

9. What does it mean to be a comparison-based sorting algorithm?

10. Give a lower bound on the running time of any comparison-based sorting algorithm.

11. What is the minimum number of comparisons needed to find the maximum in a set of n elements?

12. Give an example of a classic sorting algorithm that is $\theta(n^2)$.

13. What is a treap?

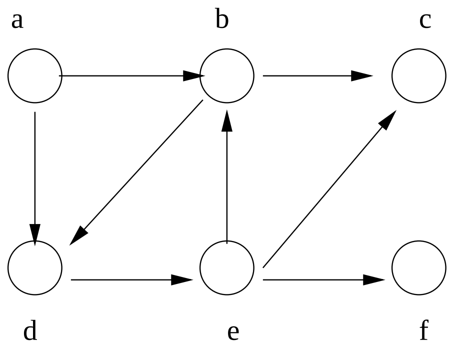
14. What is the "heap" property?

15. Give a θ-bound on the time it takes to find the median of n integers.

16. Briefly describe the median of medians rule for selecting a pivot in Quicksort.

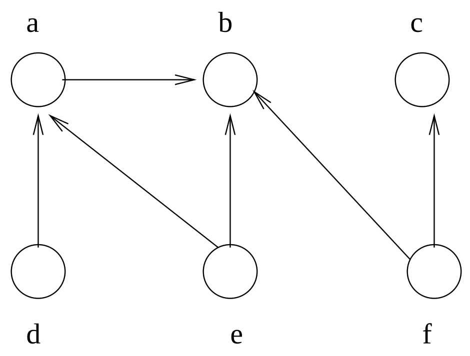
17. What is a "cut" in a graph?

18. Draw the DFS tree for the following digraph when alphabetical priority is used.



19. We run the DFS algorithm and build a timestamp table for a digraph G. There are vertices u and v in G such that $\mathrm{FA}(u) < \mathrm{LD}(v)$, where FA denotes first-arrival time and LD denotes last-departure time. List all orderings of $\mathrm{FA}(u)$, $\mathrm{LD}(u)$, $\mathrm{FA}(v)$, $\mathrm{LD}(v)$ that are possible in the timestamp table.

20. List the vertices of the following digraph in topological order.



21. What is a feasible flow in a flow network? (There are two conditions you should mention.)

22. What does it mean for an algorithm to be "greedy"?

23. Give a yes-instance and a no-instance of Vertex Cover.

24. Prove that 3SAT is in NP.

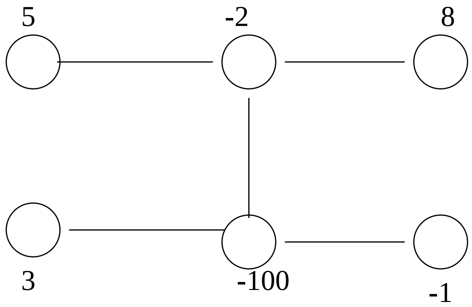
25. What does it mean for a problem to be NP-complete? (There are two conditions you should mention.)

Part B: The second part of the exam consists of three problems. Each problem is worth 25 points.

1. You are given a set of integers $S = \{x_1, \ldots, x_n\}$ that are semi-sorted. By semi-sorted we mean that the rank of x_i in S is between $i - \lg n$ and $i + \lg n$. How much time does it take to sort the input? More specifically, does the lower bound for comparison-based sorting hold for this type of input? You should either prove the lower bound argument still holds or give an algorithm that sorts the list in some time that breaks the bound.

2. Let T be a vertex-weighted tree. An independent set of T is a subset of vertices W such that no pair of vertices in W have an edge of T. The weight of an independent set W is 1 if W is empty and otherwise it is the product of the weights of the vertices in W.

   (a) Find a maximum weight independent set of the following tree.

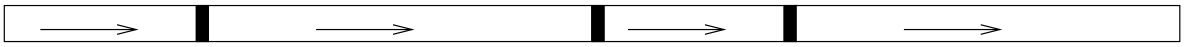


   (b) Design a recursive algorithm for this problem. Clearly define any terms/variable names you use. You should explain your main ideas in English, not pseudo-code. Be sure to describe any base cases your algorithm needs. What is the running time of your algorithm?

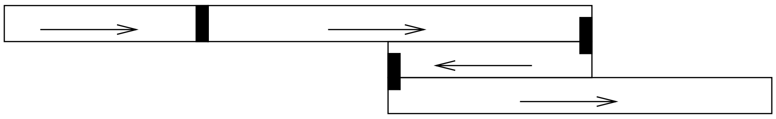
(c) Design a dynamic programming algorithm that solves this problem efficiently. Analyze the running time of your algorithm.

3. A carpenter's ruler is shown in the figure below. 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. The following figures shows the ruler folded at the second and third hinge.

No folds:



Folded at second and third hinge:



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

Carpenter's Ruler Problem (CRP):

Input: Integer lengths of the segments of a carpenters 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) Enumerate all possible folded lengths of the ruler in Figure 1.

(b) Give a yes-instance and a no-instance of the CRP.

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

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

i. Let $S = \{x_1, \ldots, x_n\}$ be an instance of Partition. Describe your algorithm for producing the associated instance of CRP.

ii. Suppose the instance S is a yes-instance of Partition. Prove that the instance of CRP that is produced by your reduction is also a yes-instance.

iii. Suppose the instance of CRP produced by your reduction is a yes-instance. Prove that the original instance of partition is also a yes-instance.