15-451 Algorithms, Spring 2016
Final Exam (Version B)

Name: ________________________ **Andrew ID:** ________________________

There are 6 problems, for a total of 180 points. You have 180 minutes. Closed book. Two sheets of notes allowed. No calculators, cell phones, internet, texting, tweeting, etc.

Problem	1	2	3	4	5	6	total
Score							
Out of	40	35	25	30	30	20	180

1. Short-answer questions. (40 pts)

(a) If you have given us feedback using the FCEs and the TA review forms, please write "I have given feedback" below and sign.

____

(b) Suppose you have the option of running one of three algorithms to solve a given problem. On an input of size n (assume $T(1) = 1$):

- Algorithm A breaks it into 4 pieces of size $n/4$, recursively solves each piece, and then combines the solutions in time n.

  The runtime is $\Theta($________________________$)$

- Algorithm B breaks it into 2 pieces of size $n/2$, recursively solves each piece, and then combines the solutions in time $n \log_2 n$.

  The runtime is $\Theta($________________________$)$

- Algorithm C breaks it into $\sqrt{n}$ pieces of size $\sqrt{n}$, recursively solves each piece, and then combines the solutions in time n.

  The runtime is $\Theta($________________________$)$

The fastest algorithm is ______ and the slowest is ______.

(c) Given an unsorted array of n distinct elements, you want to find this set of $\lg n$ elements: those at positions $1, 2, 4, 8, 16, \ldots, n/2$ in the sorted order. (The element at position 1 in the sorted order is the minimum, the one in position 2 is the 2nd smallest, $\cdots$, and the one in position n is the largest.) Assume n is a power of 2. How fast can you find all these $\lg n$ elements?

$$\Theta(\log n) \qquad \Theta(\sqrt{n}) \qquad \Theta(n) \qquad \Theta(n \log n) \qquad \Theta(n^2)$$

(d) You want a universal hash family from $U = \{0,1\}^n$ to $T = \{0,1\}^m$.

(a) Briefly describe a construction for such a hash family?

(b) How many hash functions are in this hash family?

____

(c) Is the hash family you constructed above also 2-universal? (Briefly explain why or why not.)

(e) True or False:

T F There are problems in P that we do not currently know are in NP.

T F There are problems in NP that we don't currently know are in P.

T F To show that $P = NP$, it suffices to show that some NP-complete problem is in P.

T F To show that $P = NP$, it suffices to show that some problem in NP is in P.

T F To show that $P = NP$, it suffices to show that the vertex cover problem has a 1.1-approximation.

T F If $P = NP$, then the CONNECTIVITY problem that says *YES* if the input is a connected graphs and *NO* for non-connected graphs is NP-complete.

(f) Choose the best upper bound on the number of prime divisors of a number n from the options below.

$\Theta(1)$ $\Theta(\log n)$ $\Theta(\sqrt{n})$ $\Theta(n)$ $\Theta(n \log n)$ $\Theta(2^n)$

(g) If you want the interval $[1 \ldots M]$ to contain at least n primes, what is the smallest value of M that would suffice.

$\Theta(1)$ $\Theta(\log n)$ $\Theta(\sqrt{n})$ $\Theta(n)$ $\Theta(n \log n)$ $\Theta(2^n)$

(h) In a planar triangulated graph with n vertices, what is the average degree?

____

(i) When incrementing a number (expressed in *decimal*) from 0 to n, what is a tight upper bound (independent of n) on amortized number of digits that change on each increment?

____

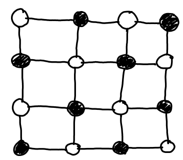
(j) True or False:

T F Since the runtime of perceptron depends on $R^2 = \max_i \|\mathbf{a}_i\|^2$, we scale the runtime by $\frac{1}{4}$ if we scale the input vectors by $\frac{1}{2}$.

T F We do not know how to find a hyperplane that separates positively labeled points in $\mathbf{R}^n$ from negatively labeled ones in polynomial time.

2. More short-answer questions. (35 pts)

(a) Let G_n be an $n \times n$ grid graph. The graph G_4 is shown below. The partitioning into hollow and solid vertices shows that this graph is bipartite.



Now let G'_n be a flow graph obtained from G_n by adding a new vertex s and connecting it to all the solid vertices, and a new vertex t and connecting it to all the hollow vertices. Each edge (in each direction) has a capacity of 1.

In G'_n, what is the maximum flow from s to t as a function of n?

____

(b) Consider a primal LP: maximize $\mathbf{c}^T\mathbf{x}$ subject to $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq 0$, where A is an $m \times n$ matrix. Suppose the optimal value of this LP is $v \in \mathbb{R}$, which is finite.

T F The optimum value of the dual is exactly v.

T F The value of every dual solution is at least v.

The number of constraints in the dual is __________.

The number of variables in the dual is __________.

T F For each primal solution $\mathbf{x}$ there is a dual solution $\mathbf{y}$ such that the value of the primal LP on $\mathbf{x}$ equals the value of the dual LP on $\mathbf{y}$.

(c) The department wants to assign offices numbered a, b, c to three TAs: Aakash, Bojian, and Chris. Each gets one office. Aakash's valuations for these offices are $\{7, 1, 9\}$, Bojian's for these offices are $\{4, 6, 8\}$, Chris' valations are $\{7, 6, 7\}$.

1. What is the welfare-maximizing perfect matching? ____________________
2. What is the VCG price for Aakash? __________
3. What is Aakash's utility for the room allocated to him by VCG? __________

(d) Write down a quadratic polynomial $P(x)$ that has

$$P(1) = 5, \quad P(3) = -8, \quad P(11) = 21.$$

(You don't have to simplify it.)

(e) For each of the following problems, label it with the paradigm presented in class to solve it. If the problem was not presented in class, then use your judgement on which paradigm is most appropriate for the problem.

The paradigms to use are *sweep-line, sweep-angle, divide and conquer, random incremental,* and *binary search.* (You may use these each zero or more times.)

(i) Solving 2-D linear programming with m constraints in $O(m)$ time.

____

(ii) Given n points in the plane and a radius r find the circle of radius r that contains the most points. The running time is $O(n^2 \log n)$.

____

(iii) Given a collection of n line segments in the plane, determine in $O(n \log n)$ time if any of them intersect.

____

(iv) Given n points in the plane find the closest pair in $O(n)$ time.

____

(v) Given n points in the plane find the closest pair in *deterministic* $O(n \log n)$ time.

____

(vi) Given a collection of n rectangles in the plane, determine the area of the union of all of them in $O(n \log n)$ time.

____

3. Dynamic Tiles (25 pts)

You're given a $1 \times N$ rectangle (height 1 and width N) and an unlimited supply of tiles of height 1 and width taken from a set of positive numbers $B = \{b_1, b_2 \ldots, b_k\}$, where $b_i < b_{i+1}$. A tiling is *valid* if the widths sum to N, and no two tiles of the same width are adjacent. The task is to count the number of valid tilings.

> **Example:** If $B = \{2, 4\}$, and $N = 6$ then the number of valid tilings is 2. ($[2, 4]$ and $[4, 2]$). If $B = \{1, 2, 3\}$ and $N = 6$ then the number is 8. (Six permutations of $[1, 2, 3]$, and also $[1, 2, 1, 2]$ and $[2, 1, 2, 1]$.)

Let $f(n, i)$ be the number of valid tilings of a $1 \times n$ rectangle in which the rightmost tile is $1 \times b_i$. If $i = 0$ it means the rectangle is tiled using zero tiles, which of course is only possible if $n = 0$.

(a) Write a formula for the number of valid tilings of the $1 \times N$ ($N > 0$) rectangle in terms of $f(\cdot, \cdot)$. In the box below, put the range of a variable j to be summed over.

$$\# \text{ of valid tilings } = \sum_{\boxed{}} \underline{\hspace{6cm}}.$$

(b) Fill in the blanks in the following recurrence for $f(n, i)$.

$$f(n,i) = \begin{cases} 0 & \text{if } n < 0 \\ 0 & \text{if } n > 0 \text{ and } i = 0 \\ \underline{\hspace{6cm}} & \text{if } n = 0 \text{ and } i \neq 0 \\ \underline{\hspace{6cm}} & \text{if } n = 0 \text{ and } i = 0 \\ \sum_{\boxed{}} \underline{\hspace{4cm}} & \text{otherwise} \end{cases}$$

(c) After memoizing this recurrence, what is the worst-case runtime (in terms of N and k) to compute $f(n, i)$ for all $1 \le n \le N$ and $1 \le i \le k$?

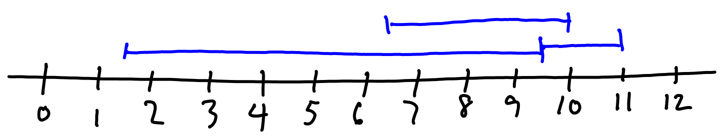
$$O(\underline{\hspace{4cm}})$$

4. Interval Coverage (30 pts)

You're given a list of *closed* intervals with endpoints on the real line. Let the intervals be $[a_1, b_1], [a_2, b_2], \ldots [a_n, b_n]$. For all $1 \le i \le n$ we have $a_i < b_i$.

The *coverage* of a point on the real line is the number of intervals that contain it. The problem is to find the maximum coverage of any point on the real line.

> For example, if the intervals are $[1.5, 9.5], [9.5, 11], [6.5, 10]$ then the coverage of 9.5 is 3, and this is the maximum. This is shown in the figure below, where the intervals are spaced out vertically for clarity.



In the following parts you will develop two different $O(n \log n)$ algorithms for this problem.

Algorithm A. First, give a sweep-line algorithm for this problem. (Read all parts before beginning to write your answer.)

(a) What types of events can occur? How do you order them? Show the events (in their proper order) for the example above.

(b) What information is being kept as you sweep along?

(c) What you do in each type of event that occurs?

Algorithm B. Now we'll develop a second algorithm for this problem. It will be based on a data structure that you learned about in this course. (Read all parts before beginning to write your answer.)

i. The first thing is to convert the real numbers corresponding to the endpoints into integers. Explain how to convert the given problem to an equivalent one (i.e., having the same maximum coverage value) in which all the endpoints are integers with values in the range $[0, 2n]$. What is the running time of this part?

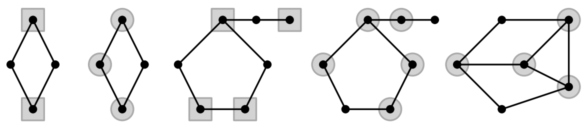
ii. List the modified set of intervals that are generated for the example above. (Reminder: the example is: $[1.5, 9.5], [9.5, 11], [6.5, 10]$)

iii. In the next part you will process these intervals via a data structure. What is this data stucture called, and what API do you need it to support?

iv. Now explain the algorithm using the data structure above. Demonstrate it by writing the sequence of calls to the API that will be made when it is run on the above example.

5. Only Connect! (30 pts)

Given a graph $G = (V, E)$, a *connected cover* is a subset S of vertices such that (a) S is a vertex cover of G, and (b) S is connected in the graph.



Example: The marked boxes in the first and third graphs are vertex covers but *not* connected covers. The marked circles in the other three graphs are connected covers, since (a) they are vertex covers, and (b) you can go between any two marked vertices using just marked vertices.

The ConnCover problem takes as input a graph $G = (V, E)$ and a positive integer K, and outputs Yes exactly when G has a connected cover of size at most K.

Show that the ConnCover problem is **NP**-complete. Specifically, fill in the blanks in the proof below.

ConnCover is in **NP** because a verifier can efficiently check a proposed solution as follows:

____

To prove that ConnCover is **NP**-hard, we can reduce from

$Q =$ ____________________ to $Q' =$ ____________________.

Specifically, given an instance I of Q we create an instance $f(I)$ of Q' as follows:

____

____

____

Now we show that

$$I \text{ is a YES-instance of } Q \iff f(I) \text{ is a YES-instance of } Q'$$

as follows:

____

____

____

Finally, time taken to write down $f(I)$ is ____________. QED.

6. Approximating Connected Covers (20 pts)

We now give a factor-3 approximation to the *connected cover* problem. Let K_{CC} be the size of the optimal connected cover in a connected graph G.

(a) Suppose K_{VC} is the size of the optimal vertex cover in G. Relate K_{VC} to K_{CC} (by filling the blank with some relation in $\{=, \leq, \geq\}$).

$$K_{VC} \quad \underline{\qquad\quad} \quad K_{CC}.$$

(b) Let C be a 2-approximate vertex cover of G (without worrying about connectivity) found using the greedy algorithm from lecture. (I.e, the one that repeatedly picks both endpoints of uncovered edges.) Argue that $|C| \leq 2K_{CC}$.

(c) Show that for any partition of the vertex cover C into two sets A, B, there exists a path of length at most 2 that connects some vertex $a \in A$ to some vertex $b \in B$.

(d) Use the fact above to give an algorithm to find a set of vertices C' such that $C \cup C'$ is a connected cover of G of size at most $3K_{CC}$. (An algorithm that achieves $4K_{CC}$ will get full credit.)