

Final Exam

- Do not open this exam booklet until you are directed to do so. Read all the instructions first.
- The exam contains 6 problems, each with multiple parts. You have 180 minutes to earn 180 points.
- This exam booklet contains 17 pages, including this one, and two sheets of scratch paper which can be detached.
- This exam is closed book. You may use 3 double-sided Letter ($8\frac{1}{2}'' \times 11''$) or A4 crib sheets. No calculators or programmable devices are permitted. Cell phones must be put away.
- Write your solutions in the space provided. If you run out of space, continue your answer on the back of the same sheet and make a notation on the front of the sheet.
- Do not waste time deriving facts that we have studied. It is sufficient to cite known results.
- Do not spend too much time on any one problem. Generally, a problem's point value is an indication of how many minutes to spend on it.
- Show your work, as partial credit will be given. You will be graded not only on the correctness of your answer, but also on the clarity with which you express it. Please be neat.
- Good luck!

Problem	Title	Points	Parts	Grade	Initials
1	True or False	40	8		
2	Short Answer	20	4		
3	Estate Showing	30	3		
4	Credit Card Optimization	30	3		
5	Well-Separated Clusters	30	3		
6	Looking for a Bacterium	30	2		
Total		180			

Name: _____

Problem 1. True/False/Justify [40 points] (8 parts)

Circle **T** or **F** for each of the following statements to indicate whether the statement is true or false, and briefly explain why. Your justification is worth more points than your true-or-false designation.

(a) **T F** If a problem has an algorithm that is correct for $2/3$ fraction of the inputs, then it also has an algorithm that is correct for 99.9% of the inputs.

(b) **T F** If a problem has a randomized algorithm that runs in time t and returns the correct answer with probability at least $2/3$, then the problem also has a deterministic algorithm that runs in time t and always returns the correct answer.

- (c) **T F** A perfect hash table that is already built requires 2 hash functions, one for the first level hash, and another for the second level hash.
- (d) **T F** If Φ is a potential function associated with a data structure S , then 2Φ is also a potential function that can be associated with S . Moreover, the amortized running time of each operation with respect to 2Φ is at most twice the amortized running time of the operation with respect to Φ .
- (e) **T F** If we use van Emde Boas (vEB) to sort n elements, it is possible to achieve $O(n \lg \lg n)$ running time. Thus, whenever we need to use traditional $O(n \lg n)$ sorting, we can replace it with vEB sort and achieve a better asymptotic running time (ignore setup time).

(f) **T F** The “union-by-rank” and “path-compression” heuristics do not improve the running time of MAKE-SET in the union-find data structure.

(g) **T F** There is an NP-hard problem with a known polynomial time randomized algorithm that returns the correct answer with probability $2/3$ on all inputs.

(h) **T F** Every special case of an NP-hard problem is also NP-hard.

Problem 2. Short Answer [20 points] (4 parts)

(a) Recall the forest-of-trees solution to the disjoint-set problem. Suppose that the only heuristic we use is a variant of the union-by-rank heuristic: when merging two roots u and v , we compare the number of descendants (rather than the rank), and make u a child of v if and only if u has fewer descendants. Is this asymptotically worse than the original union-by-rank heuristic? Explain why or why not.

(b) Suppose we apply Huffman coding to an alphabet of size 4, and the resulting tree is a perfectly balanced binary tree (one root with two children, each of which has two children of its own). Find the maximum frequency of any letter.

- (c) In lecture, we saw min-radius clustering, in which the goal was to pick a subset of k points such that each point formed the center of a cluster of radius r . Suppose instead that the center of the cluster can be a point not in the original set.

Give an example set of points where it is possible to find k clusters of radius r centered around arbitrary points, but impossible to find k clusters of radius r centered around points in the set.

- (d) Consider the following algorithm, which is intended to compute the shortest distance among a collection of points in the plane:

- 1 Sort all points by their y -coordinate.
- 2 **for** each point in the sorted list:
- 3 Compute the distance to the next 7 points in the list.
- 4 **return** the smallest distance found.

Give an example where this algorithm will return a distance that is not in fact the overall shortest distance.

Problem 3. Estate Showing. [30 points] (3 parts)

Trip Trillionaire is planning to give potential buyers private showings of his estate, which can be modeled as a weighted, directed graph G containing locations V connected by one-way roads E . To save time, he decides to do k of these showings at the same time, but because they were supposed to be private, he doesn't want any of his clients to see each other as they are being driven through the estate.

Trip has k grand entrances to his estate, $A = \{a_1, a_2, \dots, a_k\} \subset V$. He wants to take each of his buyers on a path through G from their starting location a_i to some ending location in $B = \{b_1, b_2, \dots, b_k\} \subset V$, where there are spectacular views of his private mountain range.

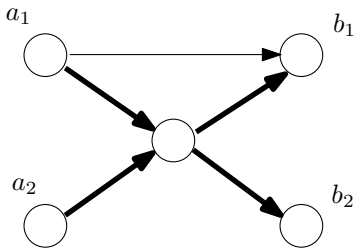
Because of your prowess with algorithms, he hires you to help him plan his buyers' visits. His goal is to find a path for each buyer i from the entrance they take, a_i , to any ending location b_j such that no two paths share any edges, and no two buyers end in the same location b_j .

- (a) Trip tells you his idea: find all-pairs shortest paths on G , and then try to select k of those shortest paths $a_i \rightsquigarrow b_j$ such that all k paths start and end at different vertices and no two paths share any edges.

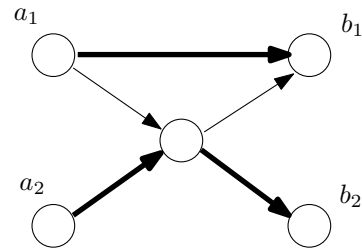
Give a graph where there exists a set of paths satisfying the requirements, but where Trip's strategy won't find it.

- (b) Rather than using shortest paths, you think that perhaps you can formulate this as a flow problem. Find an algorithm to find k paths $a_i \rightsquigarrow b_j$ that start and end at different vertices and that share no edges, and briefly justify the correctness and running time of your algorithm.

- (c) Trip, after trying out the paths found by your algorithm, realizes that making sure the k paths don't share edges isn't enough: it's possible that some paths will share vertices, and his buyers might run into each other where their paths intersect.



$a_1 \rightsquigarrow b_1$ and $a_2 \rightsquigarrow b_2$ share a vertex



$a_1 \rightsquigarrow b_1$ and $a_2 \rightsquigarrow b_2$ share neither vertices nor edges

Modify your algorithm to find k paths $a_i \rightsquigarrow b_j$ that start and end in different locations, and that share neither vertices nor edges.

Problem 4. Credit Card Optimization [30 points] (3 parts)

Zelda Zillionaire is planning on making a sequence of purchases with costs x_1, \dots, x_n . Ideally, she would like to make all of these purchases on one of her many credit cards. Each credit card has credit limit ℓ . Zelda wants to minimize the number of credit cards that she needs to use to make these purchases, without exceeding the credit limit on any card. More formally, she wants to know the smallest k such that there exists $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, k\}$ assigning each purchase i to a credit card j , where π must satisfy the following constraint:

$$\forall j \in \{1, \dots, k\} : \sum_{\substack{i \in \{1, \dots, n\} \\ \text{s.t. } \pi(i)=j}} x_i \leq \ell.$$

Zelda is thinking of using the following algorithm to distribute her purchases:

```

1  Create an empty list of credit cards  $L$ .
2  for each purchase  $x_i$ :
3      for each credit card  $j \leq |L|$ :
4          if  $L[j] + x_i \leq \ell$ :
5              Purchase  $x_i$  with credit card  $j$ .
6              Set the balance for card  $j$  to  $L[j] = L[j] + x_i$ .
7              Skip to the next purchase.
8          if no existing credit card has enough credit left:
9              Purchase  $x_i$  with a new credit card.
10             Append a new credit card to  $L$ .
11             Set the balance of the new credit card to  $x_i$ .
12 return  $k = |L|$ 

```

(a) Give an example where Zelda's algorithm will not use the optimal number of credit cards.

(b) Show that Zelda's algorithm gives a 2-approximation for the number of credit cards.

HINT: Let OPT be the optimal number of credit cards. In order for the algorithm to add a new credit card $j > OPT$, it must reach x_i such that $x_i + \min_{j' \in \{1, \dots, OPT\}} L[j'] > \ell$.

It will then set $L[j] = x_i$.

(c) Show that minimizing the number of credit cards used is NP-hard.

Problem 5. Well-Separated Clusters [30 points] (3 parts)

Let $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$ be a set of n points in the plane. You have been told that there is some k such that the points can be divided into k clusters with the following properties:

1. Each cluster has radius r .
2. The center of each cluster has distance at least $5r$ from the center of any other cluster.
3. Each cluster contains at least ϵn elements.

Your goal is to use this information to construct an algorithm for computing the number of clusters.

- (a) Give an algorithm that takes as input r and ϵ , and outputs the exact value of k .

(b) Given a particular cluster C_i , if you sample t points uniformly at random (with replacement), give an upper bound in terms of ϵ for the probability that none of these points are in C_i .

(c) Give a sublinear-time algorithm that takes as input r and ϵ , and outputs a number of clusters k that is correct with probability at least $2/3$.

Problem 6. Looking for a Bacterium [30 points] (2 parts)

Imagine you want to find a bacterium in a one dimensional region divided into n $1\text{-}\mu\text{m}$ regions. We want to find in which $1\text{-}\mu\text{m}$ region the bacterium lives. A microscope of resolution r lets you query regions 1 to r , $r + 1$ to $2r$, etc. and tells you whether the bacteria is inside that region. Each time we operate a microscope to query one region, we need to pay (n/r) dollars for electricity (microscopes with more precise resolution take more electricity). We also need to pay an n -dollar fixed cost for each type of microscope we decide to use (independent of r).

- (a) Suppose that you decide to purchase $\ell = \lg n$ microscopes, with resolutions r_1, \dots, r_ℓ such that $r_i = 2^{i-1}$. Give an algorithm for using these microscopes to find the bacterium, and analyze the combined cost of the queries and microscopes.

- (b) Give a set of microscopes and an algorithm to find the bacterium with total cost $O(n \lg \lg n)$.

SCRATCH PAPER

SCRATCH PAPER

MIT OpenCourseWare
<http://ocw.mit.edu>

6.046J / 18.410J Design and Analysis of Algorithms
Spring 2012

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.