CS 473: Algorithms, Fall 2018

Midterm II: November 13, 2018, 7pm-9pm, DCL 1310

Version: **1.0**

Name:	
Netid:	

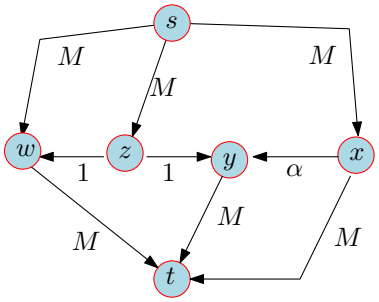
- This is a closed-book, closed-notes, open-brain exam. If you brought anything with you besides writing instruments and your **handwritten** $8\frac{1}{2}'' \times 11''$ cheat sheet, please leave it at the front of the classroom.
- Please **print** your name, and netid in the boxes above. Print your name at the top of every page (in case the staple falls out!).
- **You should answer all the questions on the exam.**
- The last few pages of this booklet are blank. Use that for a scratch paper. Please let us know if you need more paper.
- If your cheat sheet is not hand written by yourself, or it is photocopied, please do not use it and leave it in front of the classroom.
- Submit your cheat sheet together with your exam. An exam without your cheat sheet attached to it will not be checked or even graded.
- If you are NOT using a cheat sheet you should indicate it in large friendly letters on this page.
- There are 4 questions on the exam. Each question is worth 25 points.
- Answers containing *only* the expression: "I don't know", will get 25% of the points of the question. If you write anything else, it would be ignored. Overall, points given for "I don't know" will not exceed 10 points for the whole exam.
- Write your exam using a pen (please do not use an invisible ink or a pencil).
- Time limit: 110 minutes.
- Relax.

I feel like a spinning top for a dreidel
The spinning don't stop when you leave the cradle
You just slow down
Round and around this world you go
Spinning through lives of the people you know
We all slow down
– Don McLean

1 Flow, flow, flow. (25 pts.)

[From the discussion section.]

Consider the network flow on the right. Here M is a large positive integer, and $\alpha = (\sqrt{5}-1)/2$. One can verify that (i) $\alpha < 1$, (ii) $\alpha^2 = 1-\alpha$, and (iii) $1-\alpha < \alpha$. We will be interested in running algFordFulkerson on this graph.



1.A. (2 pts.) What is the value of the maximum flow in this network?

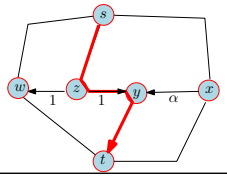
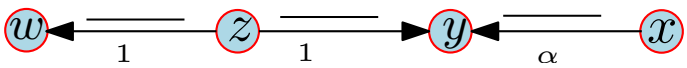
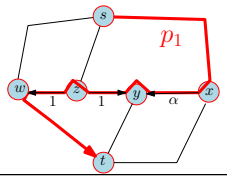
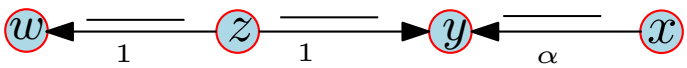
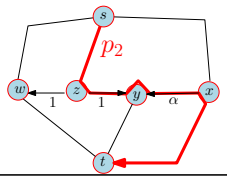
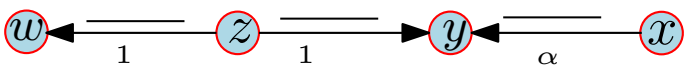
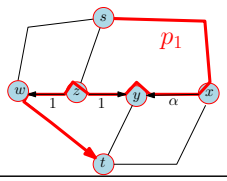
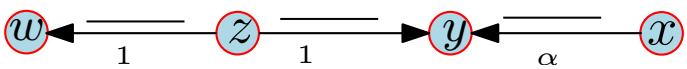
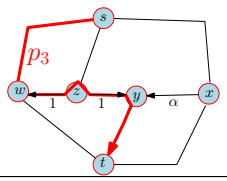
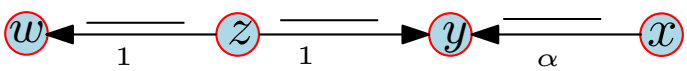
1.B. (3 pts.) Prove that $\alpha^i - \alpha^{i+1} = \alpha^{i+2}$, for all $i \geq 0$.

1.C. (10 pts.) We next specify the sequence of augmenting paths used by the algorithm. In the table below, specify the residual capacities for the designated edges after each step (the residual graph would have also other edges which we do not care about). We start here from the 0 flow.

Please simplify the numbers involved as much as possible, using the relation from (B). Hint: All the numbers you have to write below are either $0, 1$ or α^i, for some i.

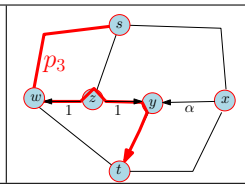
You want to be careful about the calculations - it is easy to make mistakes here.

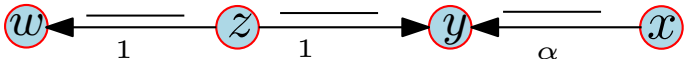
(See other side of the page for the continuation of this problem).

step	augmenting path	amount pushed	residual capacity
0.		1	
1.		α	
2.		α	
3.		α^2	
4.		α^2	

1.D. (5 PTS.) Imagine that we now repeat steps $1, 2, 3, 4$ above (i.e., we augment along p_1, p_2, p_1, p_3 repeatedly. What would be the residual capacities in the graph after the ith such repetition, starting with the results in the step 4 above?

4i.





1.E. (5 PTS.) How many iterations would it take this algorithm to terminate, if we repeat the steps in (D)?

2 Spiderman. (25 pts.)

You are given a directed graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ with n vertices and m edges. There is a set of k base stations $B \subseteq \mathsf{V}$. Each vertex in $\mathsf{V} \setminus B$ has a single message it would like to send to (any) of the base stations. Such a message can be sent along a path in G. In particular, in a *round* of the broadcasting schedule, some of the nodes sends their messages, along some paths that are specified, such that no two paths have a common edge. The problem of course is that one might need several rounds to send all the messages.

Describe an efficient approximation algorithm that minimizes the number of rounds needed till all messages get sent to the base stations.

What is the approximation quality of your algorithm? The lower the better.

What is the running time of your algorithm? (The faster, the better [approximation quality matters more than running time for this question].)

3 Fast clustering. (25 pts.)

[Building on and improving an algorithm seen in class.]

Let P be a set of $n \geq 2$ points in the plane. A ***partial cover*** D is a set of $k \geq 1$ disks $d_1, \ldots, d_k$. Here a disk d_i has radius r and it is centered at some point of P, where r is the distance between some two points of P. The partial cover D is a ***k-clustering of P*** if all the points of P are covered by the disks of D

In class, we saw an algorithm that computes, in $O(nk)$ time, a clustering of P with k disks of radius r such that $r \leq 2r_{\text{opt}}(P,k)$, where r_{opt} is the minimum radius of any cover of P by k disks of the same radius (in particular, $\forall X \subseteq P$ we have $r_{\text{opt}}(X,k) \leq r_{\text{opt}}(P,k)$).

In the following assume that $k = O(\sqrt{n})$.

3.A. (2 pts.) Give a tight upper bound on the number of partial covers, as a function of n and k.

3.B. (2 pts.) A partial cover D of P is ***bad*** if there are more than $n/2$ point of P that are not covered by the disks of D. Let $R = \{r_1, \ldots, r_t\}$ be a random sample from P of size $t = ck\lceil \log_2 n \rceil$, where c is a sufficiently large constant. Here, each r_i is chosen uniformly and independently from P.

Give an upper bound (as small as possible) on the probability that a specific bad cover D covers all the points of R (as a function of k, n, c).

3.C. (2 pts.) Let $\mathcal{B}$ be the set of all bad partial covers of P. Prove that, with high probability, no cover in $\mathcal{B}$, covers all the points of R.

3.D. (2 pts.) Let R be the random sample as described above. Let D be any k-clustering of R. Prove that D covers at least half the points of P, with high probability.

3.E. (17 pts.) Assume, that given a partial cover D with k disks, one can preprocess it, in $O(k \log k)$ time, such that given a query point p, one can decide, in $O(\log k)$ time, whether or not it is covered by the disks of D.

Describe an algorithm, as fast as possible, using the above, that computes a cover of P by $O(k \log n)$ disks of the same radius r, such that $r \leq 2r_{\text{opt}}(P,k)$. For any credit, the running time of your algorithm has to be $o(kn)$, for $k = \Omega(n^{1/4})$.

4 Hitting set. (25 pts.)

[Similar stuff seen in class.]

Let $(P, \mathcal{F})$ be a given set system – that is, P is a set of n elements, and $\mathcal{F}$ is a collection of m subsets of P. Describe an *efficient* approximation algorithm that computes a subset $X \subseteq P$, as small as possible, such that for any set $S \in \mathcal{F}$, we have that $X \cap S \neq \emptyset$.

Prove the quality of approximation of your algorithm (the quality of approximation has to be as good as possible). What is the running time of your algorithm.

*** Scratch paper - do not detach from the booklet ***

*** Scratch paper - do not detach from the booklet ***