NAME: ____________________

CMPSCI 611
Advanced Algorithms
Midterm Exam Fall 2020 Draft Solutions

A. McGregor

8pm, 11/5/20 to 8pm, 11/6/20

DIRECTIONS:

- **Honesty Policy:** No communicating with anyone (except the instructor and TAs via private posts on Piazza) about the exam during the 24 hours the exam is open. You are not allowed to use any resources except from the slides and scribed notes linked in Moodle.
- You can either print the exam and handwrite your answers on the pages or type the answers on separate pages. It should be possible to answer each question using a single page. Most parts of each question can be answered in a few sentences. Remember that the best answers are those that are clear and concise (and of course correct).
- Once you are finished submit your solutions in Gradescope by 8pm Friday. If you are using a camera to "scan" your answers, please take care to make the picture as legible as possible and remember to tag each uploaded page with the question it relates to.

1	/10
2	/10
3	/10
4	/10
5	/10
Total	/50

Question 1. For each of the following statements, indicate whether they are TRUE or FALSE by circling the appropriate answer. No justification is required.

1. There is a polynomial time algorithm for finding the minimum weight spanning tree of an edge-weighted graph.
   Answer: TRUE

2. If $E = \{a, b, c\}$ and $\mathcal{I} = \{\{\}, \{a\}, \{c\}\}$ then $(E, \mathcal{I})$ is a matroid.
   Answer: TRUE

3. If $T(n) = 3T(n-1)$ for $n \geq 2$ and $T(1) = 1$ then $T(n) = O(n^3)$.
   Answer: FALSE. $T(n) = 3^{n-1}$ and this is not $O(n^3)$.

4. Let $f(x)$ and $g(x)$ are polynomials of degree at most d such that $f(0) = g(0)$ but $f(1) \neq g(1)$. Then if we pick $r \in \{1, 2, \ldots t\}$ uniformly at random, $\Pr[f(r) = g(r)] \leq (d-1)/t$.
   Answer: TRUE. Since $f(1) \neq g(1)$, we know $f(x) - g(x)$ is a non-zero polynomial of degree at most d. Hence, it has at most d roots, one of which is $x = 0$. Hence, at most $(d-1)$ of the possible values for r are roots.

5. It is possible to sort n values with $O(n \log n)$ pairwise comparisons.
   Answer: TRUE. For example, Merge-Sort only requires $O(n \log n)$ comparisons.

Question 2. Consider the following adjacency matrix

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

and let $B = A^2$. For a matrix M, let $M[i, j]$ denote the entry in the ith row and jth column. For example, $A[1, 1] = 0$ and $A[1, 2] = 1$.

1. What is the diameter of the graph corresponding to A?
   Answer: 3. The graph is a path with 3 edges.

2. What is the size of the minimum vertex cover of the graph corresponding to A?
   Answer: 2.

3. What is the value of $B[2, 4]$?
   Answer: 1. It is the dot product of the 2nd row of A with the 4th column of A.

For the rest of the question, we consider an arbitrary adjacency matrix C corresponding to a graph $G = (V, E)$ with n nodes and m edges.

4. Let $G' = (V, E')$ be the graph where there is an edge between a pair of nodes iff there is a path of length *exactly* two between those nodes in G. We don't consider path of length two that starts and ends at the same node in this definition. Explain how to compute the adjacency matrix of G' given C. You don't need to explain how to square a matrix.

Answer: Compute C^2. Note $C^2[i,j]$ is the number of length two paths between node i and j. Hence, if D is the adjacency matrix of G', $D[i,j] = \min(1, C^2[i,j])$ if all $i \neq j$ and $D[i,i] = 0$ for all i.

5. The trace of a matrix is the sum of the entries on the diagonal. What is the value of trace of C^2 in terms of n or m or some combination of both?

   Answer: $C^2[i,i] = \sum_j C[i,j]C[j,i] = |\{j : (i,j) \in E\}|$ is the degree of node i. So the trace is the sum of the degrees and this is $2m$.

Question 3. A candidate wants to become president in a country with n states. Winning the ith state gives the politician v_i points. The candidate needs a total of at least k points in order to become president. Let $D[i,j]$ be the number of subsets $S \subseteq \{1, 2, \ldots, i\}$ such that $\sum_{s \in S} v_s \geq j$. Then $D[n,k]$ is the number of ways the candidate can win the election.

1. Design a dynamic programming algorithm to compute $D[n,k]$ that runs in time polynomial in k and n. Remember to prove correctness and analyze the running time.

   Answer:

   1. Let $D[0,j] \leftarrow 0$ for all $j \geq 1$
   2. For $i = 1$ to n:
      2.1. For $j = 1$ to k:
         3.1. If $v_i < j$ then $D[i,j] \leftarrow D[i-1,j] + D[i-1, j-v_i]$
         3.2. If $v_i \geq j$ then $D[i,j] \leftarrow D[i-1,j] + 2^{i-1}$
   3. Return $D[n,k]$

   Correctness follows because every $S \subseteq [i]$ with $\sum_{s \in S} v_s \geq j$ is either a subset of $[i-1]$ with total at least j or includes i and a subset of $[i-1]$ with total at least $j - v_i$. Hence, $D[i,j]$ is $D[i-1,j] + D[i-1, j-v_i]$ and note that if $j - v_j \leq 0$, then any of the 2^{i-1} subsets of $[i-1]$ has weight at least $j - v_i$. The running time is $O(nk)$.

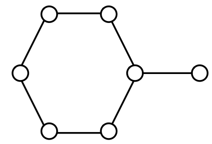
2. Suppose the candidate does not want to campaign in any state that is not worth many points. Design an algorithm that, given input $v_1, \ldots, v_n, k, t$ finds the largest value θ such that there are at least t subsets $S \subseteq \{i \in [n] : v_i \geq \theta\}$ such that $\sum_{s \in S} v_s \geq k$. The running time should be at most a factor $O(\log n)$ larger than the running time of the algorithm in Part 1.

   Answer: Let D_θ be the value returned by the algorithm in Part 1 when all elements whose value is strictly less than θ are removed. Note that D_θ is monotonically decreasing as θ increases. Therefore, we can binary search for the largest v_i such that $D_{v_i} \geq t$. To do this, we first sort the v_i. Total running time is $O(n \log n + nk \log n) = O(nk \log n)$.

3. Design an algorithm using $O(n \log n)$ time that computes the largest θ such that there exists at least one subset $S \subseteq \{i \in [n] : v_i \geq \theta\}$ such that $\sum_{s \in S} v_s \geq k$.

   Answer: In $O(n \log n)$ time, we can sort the values such that $v_1 \geq v_2 \geq \ldots \geq v_n$. In $O(n)$ time, we can compute the minimum i such that $v_1 + v_2 + \ldots + v_i \geq k$. Note that $S = \{1, 2, \ldots, i\}$ is valid subset with $\theta = v_i$. This is the max possible θ since any other valid subset S would include a state j with $v_j \leq v_i$.

Question 4. Consider a graph consisting of a cycle of length n and one extra node that is connected to one node in the cycle. Call this graph G_n. For example, G_6 looks like this



1. Assuming n is even, what are the following values as a function of n?

   Size of min cut in G_n $=$ 1 Size of max cut in G_n $=$ $n+1$

2. Consider Karger's algorithm: 1) Pick a random edge, contract this edge, and repeat until only two nodes remain. 2) Let X be the number of edges between the final two nodes. What is the exact value of $\Pr[X = 1]$ when the algorithm is run on graph G_n?

   Answer: The only way for $X = 1$ is if each random edge is from the cycle rather than the edge sticking out from the cycle. Hence, $\Pr[X = 1] = \frac{n}{n+1} \times \frac{n-1}{n} \times \frac{n-2}{n-1} \times \ldots \times \frac{2}{3} = \frac{2}{n+1}$.

3. Suppose you instead pick a random subset S of nodes where each node is included in S independently with probability $1/2$. Let Y be the number of edges that have exactly one endpoint in S. What is the exact value of $\Pr[Y = 1]$ when the algorithm is run on graph G_n?

   Answer: The only way for $Y = 1$ is if S is the set of nodes in cycle or the single node not in the cycle. Hence, $\Pr[Y = 1] = 1/2^{n+1} + 1/2^{n+1} = 1/2^n$.

4. Is Y (see above) distributed according to a binomial distribution? Explain your answer.

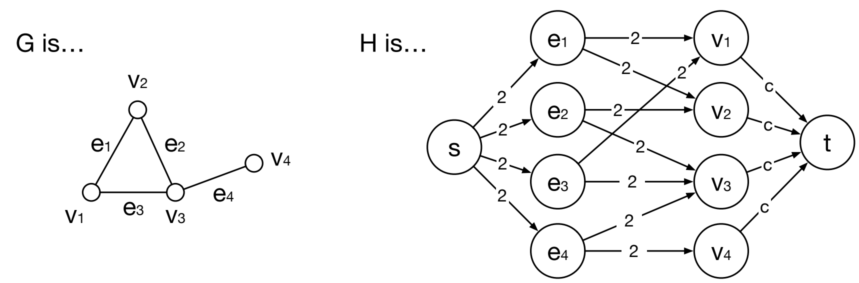
   Answer: No. Y takes values in $\{0, \ldots, n+1\}$ and $E[Y] = (n+1)/2$. So if Y is binomial it has parameters $n+1$ and $p = 1/2$. But if Y is binomial with these parameters then $\Pr[Y = 1] = (n+1)/2^{n+1}$ and this isn't equal to $1/2^n$ as computed above.

5. Given an arbitrary graph, pick $\lfloor n/\sqrt{m} \rfloor$ random nodes without replacement where m is the number of edges. What is the expected number of edges where both endpoints are picked? Prove that every graph includes an independent set of size at least $\lfloor n/\sqrt{m} \rfloor$.

   Answer: Let $k = \lfloor n/\sqrt{m} \rfloor$. The probability both endpoints of edge are picked is $\frac{k(k-1)}{n(n-1)} < (k/n)^2 \leq 1/m$ where the strict inequality assumed $m > 1$. By linearity of expected the expected number of edges where both endpoints are picked is $< m \times 1/m = 1$. Since the expectation is strictly less than 1, there must exist a subset where the number of such edges is 0. This is an independent set of the required size.

Question 5. Given a graph and a subset of nodes U, define $\gamma(U)$ to be the average degree of the nodes in U if we only consider edges which have both endpoints in U. Let $\gamma^* = \max_U \gamma(U)$.

Consider transforming a graph G with nodes $\{v_1, \ldots, v_n\}$ and edges $\{e_1, \ldots, e_m\}$ into a network flow instance H as follows. The nodes of H are $\{s, e_1, \ldots, e_m, v_1, \ldots, v_n, t\}$. There is an edge with capacity 2 from s to e_i for each $i \in [m]$. There is an edge with capacity c from v_i to t for each $i \in [n]$. Lastly, for all $i \in [m], j \in [n]$ there is an edge with capacity 2 from e_i to v_j if v_j was an endpoint of the edge e_i in G. For example,



1. In the graph G above, if $U = \{v_3, v_4\}$ what are the following values:

$$\gamma(U) = 1 \qquad \gamma^* = 2$$

2. In the network flow instance H above:

What is max s-t flow if $c = 0$? 0 What is max s-t flow if $c = \infty$? 8

3. For an arbitrary graph G with m edges, prove that if $c < \gamma^*$ then the corresponding network has max s-t flow of size $< 2m$.

Answer: Consider $U \subseteq \{v_1, \ldots, v_n\}$ and $E(U) = \{e_i :$ both end points of e_i in $U\}$. Then $\gamma(U) = 2|E(U)|/|U|$. If there is a flow of size $2m$, there must be $2|E(U)|$ units of flow entering the nodes in U. Hence, the flow out of U is at least $2|E(U)|$. Hence, we need $c|U| \geq 2|E(U)|$, i.e., $c \geq \gamma(U)$. Since this is true for all U, we need $c \geq \gamma^*$.

4. If $c \geq \gamma^*$ it can be shown that there always exists an s-t flow of size $2m$. You do not need to show this but may assume it is true. Describe and prove the correctness of an algorithm for computing γ^*. You may use algorithms analyzed in class as sub-routines.

Answer: We need to find the minimum value of c such that there is an s-t flow of at least $2m$. For any value of c, we can find the max flow using the Ford-Fulkerson with Edmonds-Karp heuristic in $O((n + m + 2)(3m + n)^2) = O((n + m)^3)$ time since there are $n + m + 2$ nodes and $3m + n$ edges in H. To find the minimum value of c we can either try all $O(mn)$ possible values of c or, more efficiently do a binary search over c. The total running time is $O((n + m)^3 \log n)$.