NAME: ____________________________

CMPSCI 611
Advanced Algorithms
Midterm Exam Fall 2020

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. I have read and understood the honesty policy on the front page of the exam. I agree to follow this policy.

   TRUE FALSE

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

   TRUE FALSE

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

   TRUE FALSE

4. If $T(n) = 3T(n-1)$ for $n \geq 2$ and $T(1) = 1$ then $T(n) = O(n^3)$.

   TRUE FALSE

5. 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, $\Pr[f(r) = g(r)] \leq (d-1)/t$.

   TRUE FALSE

6. It is possible to sort n values with $O(n \log n)$ pairwise comparisons.

   TRUE FALSE

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?

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

3. What is the value of $B[2, 4]$?

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.

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?

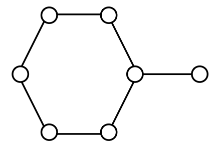
Question 3. A candidate wants to become president in a country with n states. Winning the ith state gives the politician v_i points. The candidate needs a total of at least k points in order to become president. Let $D[i, j]$ be the number of subsets $S \subseteq \{1, 2, \ldots, i\}$ such that $\sum_{s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

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.

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 $\sum_{s \in S} v_s \geq k$.

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 $=$ Size of max cut in G_n $=$

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?

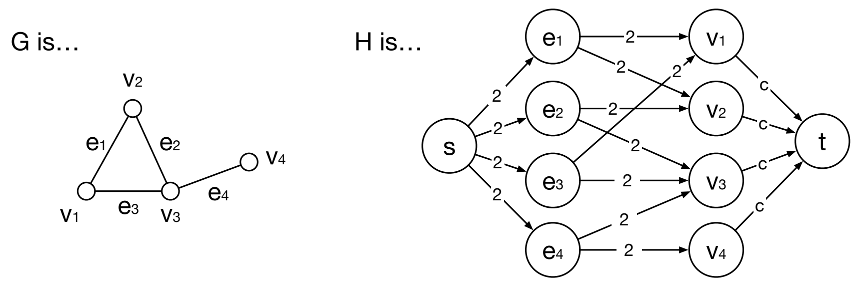
3. Suppose you instead pick a random subset S of nodes where each node is included in S independently with probability $1/2$. Let Y be the number of edges that have exactly one endpoint in S. What is the exact value of $\Pr[Y = 1]$ when the algorithm is run on graph G_n?

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

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 $\lfloor n/\sqrt{m} \rfloor$.

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:

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

2. In the network flow instance H above:

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

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 $< 2m$.

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.