NAME: ____________________________

CMPSCI 611
Advanced Algorithms
Final Exam Fall 2010

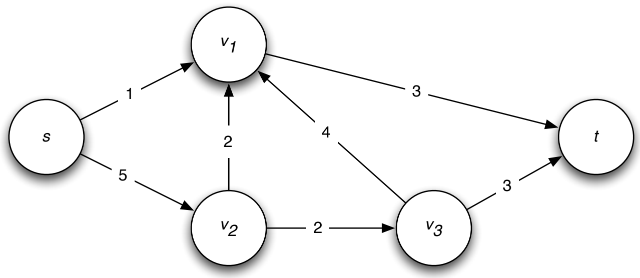
A. McGregor 13 December 2010

DIRECTIONS:

- Do not turn over the page until you are told to do so.
- This is a *closed book exam.* No communicating with other students, or looking at notes, or using electronic devices. You may ask the professor to clarify the meaning of a question but do so in a way that causes minimal disruption.
- If you finish early, you may leave early but do so as quietly as possible. The exam script should be given to the professor.
- There are five questions. All carry the same number of points but some questions may be easier than others. Don't spend too long on a problem if you're stuck – you may find that there are other easier questions.
- The front and back of the pages can be used for solutions. There are also a couple of blank pages at the end that can be used. If you are using these pages, clearly indicate which question you're answering. Further paper can be requested if required.
- The exam will finish at 3:30 pm.
- Good luck!

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

Question 1: In the first part of this question, consider the following flow network where the value of each edge represents the capacity of that edge:



1. What is the value of maximum flow possible from s to t? No proof required.
   Answer: 5.

2. What is the value of the minimum s-t cut? No proof required.
   Answer: 5.

The next part of this question concerns an arbitrary flow network. For each of the following statements, write whether they are true or false.

3. If the capacity of each edge in the network is doubled, the maximum flow is doubled.
   Answer: TRUE.

4. If the capacity of each edge that leaves the source s is increased, the maximum flow is increased.
   Answer: FALSE.

Question 2: In this question, the goal to provide examples where the suggested algorithm **does not** work. *You should specify both the correct answer and the result of the algorithm.*

1. Given a graph $G = (V, E)$ where each node v has a weight $w(v)$, we want to find the maximum weight independent set. Show a counter-example to the greedy algorithm. **Hint:** Recall that a maximum weight independent set is a subset of nodes $V' \subseteq V$ such that $\sum_{v \in V'} w(v)$ is maximized subject to the condition that $(u, v) \notin E$ for any $u, v \in V'$.
   Answer: Consider the graph with nodes u, v, w with weights $2, 3, 2$ respectively and edges between u and v and between v and w. The correct answer is $\{u, w\}$ with weight 4 whereas the greedy algorithm would just pick v and get weight 3.

2. Given a 3-CNF formula ϕ with variables $x_1, \ldots, x_n$ and an odd number of clauses, we want to determine whether ϕ is satisfiable. Consider the algorithm that assigns each x_i to TRUE if there are at least as many occurrences of the literal x_i as the literal $\bar{x}_i$. Show a satisfiable formula for which the algorithm does not find a satisfying assignment.
   Answer: Consider the formula $(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$. The algorithm sets x_1, x_2, x_3 to TRUE and this doesn't satisfy the third clause. It is possible to satisfy all clauses, e.g., x_1 and x_2 are TRUE and x_3 is FALSE.

3. Dijkstra's algorithm does not necessarily work on graphs with negative weights. Give an example of a graph with positive and negative weights (but no negative weight cycles) such that Dijkstra's algorithm doesn't work. Remember to specify the source node.

   Answer: Consider the graph consisting of nodes a, b, c with directed edges $a \to b$ of weight 3, $a \to c$ of weight 2, and $b \to c$ of weight -2. Then the shortest path from a to c has weight 1 but Dijkstra reports the weight as 2.

Question 3: A pizza restaurant chain is considering opening a series of restaurants along a highway. The n possible locations are along a straight line, and the distances of these locations from the start of the highway are, in miles and in increasing order, $m_1, m_2, \ldots, m_n$. The constraints are as follows:

- At each location, you may open at most one restaurant. A restaurant at location i makes profit p_i, where $p_i > 0$ and $i = 1, 2, \ldots, n$.
- Any two restaurants should be at least k miles apart, where k is a positive integer.

1. Give an efficient dynamic programming algorithm to find the placement of the restaurants such that the maximum profit is achieved subject to the given constraints.

   Answer: Let $A[i]$ be the maximum profit to be gained by opening some subset of the first i locations. Then $A[1] = p_1$ and if $m_i \geq k$,

$$A[i] = \max\{A[i-1], p_i + A[j]\}$$

   where j is the maximum value such that $m_j \leq m_i - k$, i.e., it's the closest previous location that is at least k miles away. If $m_i < k$ then set $A[i] = A[i-1]$. This follows because either the maximum profit doesn't require opening a restaurant at location i (in which case the max profit is $A[i-1]$) or it does (in which case the profit if $p_i + A[j]$ because we can't open any restaurants at locations $j+1, j+2, \ldots, i-1$ if we open a restaurant at location i.) We can find j in $O(\log n)$ time by doing a binary search. If we compute $A[i]$ in increasing order of i then we can compute each $A[i]$ in $O(\log n)$ time and hence can compute $A[n]$ in $O(n \log n)$ time. Note that it's actually possible to improve this running time to $O(n)$.

2. Suppose $p_1 > p_2 > \ldots > p_n$ and $m_1 = 0, m_2 = 1, \ldots, m_n = n - 1$. What is the value of the optimal solution?

   Answer: $p_1 + p_{1+k} + p_{1+2k} + \ldots p_{1+rk}$ where r in the maximum integer such that $1 + rk \leq n$.

Question 4: Given a collection of m sets $C = \{A_1, \dots, A_m\}$ and an integer d, the SET-COVER problem is to determine if there are d sets $A_{i_1}, \dots, A_{i_d}$ such that $A_{i_1} \cup A_{i_2} \cup \dots \cup A_{i_d} = \cup_{A \in C} A$.

1. Prove that SET-COVER is NP-complete. You may assume VERTEX-COVER is NP-complete.

   Answer: SET-COVER is in NP: This follows because given integers $i_1, i_2, \dots, i_d$, it is possible to compute $A_{i_1} \cup A_{i_2} \cup \dots \cup A_{i_d}$ in polynomial time and verify that it equals $\cup_{A \in C} A$. It remains to show that an NP complete problem can be reduced to SET-COVER. We will show that VERTEX-COVER $\leq_P$ SET-COVER. Given an instance (G, d) of VERTEX-COVER, for each node v define the set A_v containing all edges incident to v. Then there exists a vertex cover of size d iff there is a set cover of size d.

2. Consider a collection of m sets $C = \{A_1, \dots, A_m\}$ where every set is of size 3. Design a *simple* algorithm with approximation ratio at most 3 for the problem of finding the minimum d such that there exist d sets $A_{i_1}, \dots, A_{i_d}$ such that $A_{i_1} \cup \dots \cup A_{i_d} = \cup_{A \in C} A$. Then there exists a vertex cover of

   Answer: Let $n = |\cup_{A \in C} A|$ and note that $d \geq n/3$ since each set in the optimum solution can cover at most 3 elements. If we repeatedly pick sets such that each new set covers at least one uncovered element, we will pick at most n sets. Hence, this algorithms is a 3-approximation.

Question 5: Consider the following randomized algorithm for MAX-CUT: Given an undirected, unweighted graph $G = (V, E)$, define a cut $(A, V \setminus A)$ by independently putting each node $v \in V$ in A with probability 1/2. Let X be the number of edges that cross the cut, i.e.,

$$X = |\{(u, v) \in E : u \in A, v \notin A\}| \ .$$

1. State Chebyshev's inequality.

   Answer: For any random variable X and $t > 0$, $\mathbb{P}[|X - \mathbb{E}[X]| \geq t] \leq \mathbb{V}[X]/t^2$.

2. What is the expected value of X? Show your working.

   Answer: Let $X_{(u,v)} = 1$ if u and v are on other sides of the cut and let $X_{(u,v)} = 0$ otherwise. Note that $\mathbb{E}\left[X_{(u,v)}\right] = 1/2$. Then $X = \sum_{(u,v) \in E} X_{(u,v)}$ and so $\mathbb{E}[X] = \sum_{(u,v) \in E} \mathbb{E}\left[X_{(u,v)}\right] = m/2$.

3. What is the variance of X? Hint: Recall that if random variables $Z_1, \dots, Z_m$ are such that Z_i and Z_j are independent for any $i \neq j$, then $\mathbb{V}(Z_1 + \dots + Z_m) = \sum_{i=1}^m \mathbb{V}(Z_i)$. Remember to show that the relevant random variables are independent.

   Answer: $\mathbb{V}\left[X_{(u,v)}\right] = 1/2(1 - 1/2) = 1/4$ and hence if all $X_{(u,v)}$ are pairwise independent then $\mathbb{V}[X] = m/4$. Note that $X_{(u,v)}$ and $X_{(a,b)}$ are clearly independent if $\{a, b\} \cap \{u, v\} = \emptyset$. If the edges share an endpoint, e.g., $u = a$, then $\mathbb{P}\left[X_{(u,v)} = 1, X_{(a,b)} = 1\right] = \mathbb{P}[v \text{ and } b \text{ are on the opposite side of } u] = 1/4 = \mathbb{P}\left[X_{(u,v)} = 1\right] \mathbb{P}\left[X_{(a,b)} = 1\right]$ as required.

4. Prove a *lower bound* on $\mathbb{P}[|X - m/2| < t]$ where m is the number of edges in G.

   Answer: $\mathbb{P}[|X - m/2| < t] = 1 - \mathbb{P}[|X - \mathbb{E}[X]| \geq t] \geq 1 - \mathbb{V}[X]/t^2 = 1 - m/(4t^2)$.