CMPSCI 611 FALL '09: MIDTERM

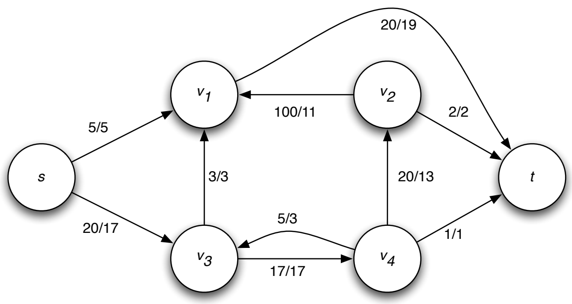
Name:

Rules:

- Do not turn over the page until you are told to do so.
- No communicating with other students, or looking at notes, or using electronic devices. You can 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. Your exam script should be given to the professor.
- There are six questions. All carry the same number of marks but some questions are easier than others. Don't spend too long on a problem if you're stuck – you may find there are other easier questions.
- The front and back of the pages can be used for solutions. There are also 4 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 from the professor if required.
- The exam will finish at 3:45 pm (according to the clock on the wall.)
- Relax...

[BLANK]

Question 1. *This question concerns network flows in the following directed graph. s is the source node and t is sink node. The label a/b on an edge indicates that the capacity of the edge is a and the flow along the edge is b.*



(1) *What's the size of the current flow? No proof required.*

(2) *What's the capacity of the cut $(\{s, v_1, v_3\}, \{v_2, v_4, t\})$? No proof required.*

(3) *Draw the residual network G_f. No proof required.*

(4) *Find an augmenting path p in G_f such that by pushing $b(p)$ (the bottleneck capacity) more flow along this path, we find a maximum flow. Write out the nodes in p. Draw the graph with the new flow. No proof required.*

(5) *State the Max-flow Min-cut theorem. By appealing to this theorem, prove that the new flow must be a maximum flow.*

[BLANK]

Question 2. *This question concerns greedy algorithms and matroids. State whether each of the following statements are true or false. No proofs or counter-examples are required for this part of the question.*

(1) $(E, \mathcal{I})$ *is a subset system if* $E = \{e_1, e_2, e_3, e_4\}$ *and* $\mathcal{I} = \{\{e_1, e_2\}, \{e_3\}, \{e_4\}, \{\}\}$.

(2) $(E, \mathcal{I})$ *is a matroid if* $E = \{e_1, e_2, e_3\}$ *and* $\mathcal{I} = \{\{e_1, e_2\}, \{e_1\}, \{e_2\}, \{e_3\}, \{\}\}$.

(3) *If* $(E, \mathcal{I})$ *is a subset system but not a matroid, then for all weight functions* $w : E \to \mathbb{R}^+$, *the greedy algorithm for the generic optimization problem fails to find a maximum solution.*

(4) *If* $(E, \mathcal{I})$ *is a matroid, and* $i, i' \in \mathcal{I}$ *are maximum with respect to a strictly positive weight function* $w : E \to \mathbb{R}^+$, *then* $|i| = |i'|$.

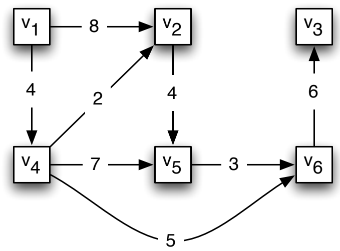
For the next parts of the question, you need to prove *things.*

(1) *For an undirected graph* $G = (V, E)$, *let* $\mathcal{I}$ *be the set of acyclic subgraphs. Prove that* $(E, \mathcal{I})$ *is a matroid. You may appeal to the Cardinality Theorem if necessary.*

(2) *You are given an undirected, connected, weighted graph* $G = (V, E)$. *All the weights* w_e *are strictly greater than 1. Give a greedy algorithm for finding the spanning tree* $T = (V, E')$ *such that* $\prod_{e \in E'} w_e$ *is maximized (i.e., note we are maximizing the product of the weights and not the sum). You must prove in detail that your algorithm is correct.*

[BLANK]

Question 3. *This first two parts of this question concern shortest paths from node v_1 in the following weighted, directed graph. The number on an edge indicates the weight of that edge.*



(1) *What's the length of the shortest weighted path from node v_1 to node v_5. No proof necessary.*

(2) *Recall that Dijkstra's algorithm maintains an array of values $[d[v_1], \ldots d[v_6]]$ in the case of a graph with 6 nodes. At the beginning of the algorithm we set $d[v_1] = 0$ and $d[v] = w(v_1, v)$ for each $(v_1, v) \in E$ and $d[v] = \infty$ otherwise. For the above graph the array of values is now $[0, 8, \infty, 4, \infty, \infty]$. During the next iteration, three of the array values are updated. What's the new array of values? No proof necessary.*

In the last part of this question, we consider an arbitrary directed graph $G = (V, E)$ with positive edge weights and a special node $v^* \in V$. We assume that the graph G is strongly connected, i.e., that for any two nodes $u, v \in V$, there is a directed path from u to v.

(3) *Give an efficient algorithm for finding shortest paths between all pairs of nodes, with the one restriction that these paths must all pass through v^*. You may express your answer in terms of any of $F(n, m)$, $D(n, m)$, or $S(n, m)$ where these are the running times of the Floyd-Warshall, Dijkstra, and Seidel algorithm respectively when ran a graph with n nodes and m edges. Prove the correctness (in detail) and claimed running time of your algorithm.*

[BLANK]

Question 4. *Consider the following game. A "dealer" produces a sequence $s_1, \ldots, s_n$ of "cards" face up, where each card s_i has a value v_i. Then two players, Alice and Bob, take turns picking a card from the sequence, but can only pick the first or the last card of the (remaining) sequence. Alice and Bob are each trying to maximize the total value of the cards they collect. Assume n is even and that Alice goes first.*

For example, suppose $n = 4$ and $v_1 = 5, v_2 = 3, v_3 = 8, v_4 = 2$. Suppose Alice first picks s_1, then Bob picks s_4, then Alice picks s_3, and then Bob picks s_2. In this case, Alice scores $5 + 8 = 13$.

(1) *Show a sequence of cards such that it is not optimal for Alice to start by picking up the available card of larger value.*

(2) *Give an $O(n^2)$ algorithm to compute an optimal strategy for Alice. Given the initial sequence, your algorithm should precompute in $O(n^2)$ time some information, and then Alice should be able to make each move optimally in $O(1)$ time by looking up the precomputed information.*

[BLANK]

Question 5. *This question involves a special type of matrices called* Hadamard matrices. *The Hadamard matrices* $H_0, H_1, H_2, \ldots$ *are defined as follows:* H_0 *is the* 1×1 *matrix* $[1]$ *and for* $k > 0$, H_k *is the* $2^k \times 2^k$ *matrix*

$$H_k = \left[\begin{array}{c|c} H_{k-1} & H_{k-1} \\ \hline H_{k-1} & -H_{k-1} \end{array}\right]$$

For example $H_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$.

(1) *Write out the matrix* H_2 *and for the vector* $v = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 2 \end{bmatrix}$, *compute* $H_2 v$.

(2) *Show that if* v *is a column vector of length* $n = 2^k$, *then the matrix-vector product* $H_k v$ *can be calculated in* $O(n \log n)$ *operations. To get full marks you must write out an algorithm, prove it is correct, and prove the running time. You may assume that all the numbers involved are small enough that basic arithmetic operations like addition and multiplication take unit time.*

[BLANK]

Question 6. *Let A and B be two sorted lists each containing n elements. For simplicity assume that n is a power of 2 and that all $2n$ elements are distinct. Give an $O(\log n)$ comparison-based algorithm for finding the nth smallest element in $A \cup B$. You need to prove that the algorithm is correct and that the running time is as claimed.*

[BLANK]

[BLANK]

[BLANK]

[BLANK]

[BLANK]