NAME: ____________________________

# CMPSCI 611
# Advanced Algorithms
# Midterm Exam Fall 2010

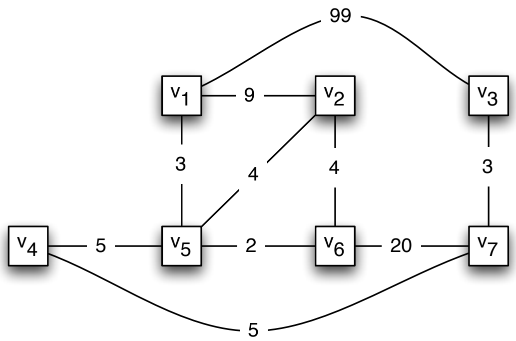
A. McGregor — 19 October 2010

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 marks 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 3:45 pm.

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

**Question 1:** In the first part of this question, consider the following undirected graph where the value of each edge represents the length of that edge:




1. What is the total length of the shortest path between $v_1$ and $v_3$?

2. What is the total length of the edges in a minimum spanning tree?

The next part of this question concerns an arbitrary weighted, undirected graph. For any two nodes $u$ and $v$ let $\delta_G(u, v)$ denote the length of the shortest path between $u$ and $v$ in the graph $G$. For each of the following statements, write whether they are true or false (no proofs required although including good reasoning *may* get partial credit even if you get the final answer wrong):

3. $\delta_T(u, v) = \delta_G(u, v)$ if $T$ is a minimum spanning tree of $G$.

4. $\delta_G(u, v) \leq \delta_G(u, w)$ implies $\delta_{G'}(u, v) \leq \delta_{G'}(u, w)$ where $G'$ is the graph formed by adding 1 to each of the edge lengths in $G$.

5. $\delta_G(u, v) \leq \delta_G(u, w)$ implies $\delta_{G''}(u, v) \leq \delta_{G''}(u, w)$ where $G''$ is the graph formed by doubling each of the edge lengths in $G$.

**Question 2:** This question is about the subset system $(E, \mathcal{I})$ where:

$$\begin{aligned} E &= \{a, b, c, d\} \\ \mathcal{I} &= \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, b\}, \{a, c\}, \{b, c\}, \{c, d\}, \{a, b, c\}\} \end{aligned}$$

1. List all the maximal subsets in $\mathcal{I}$. Why can you conclude that $(E, \mathcal{I})$ is not a matroid?

2. Consider the weighting function $w(a) = 1, w(b) = 2, w(c) = 3$, and $w(d) = 4$. What solution is returned by the greedy algorithm? How does this compare to the optimal solution?

3. Specify a weight function $w$ on $E$ such that the greedy algorithm doesn't return an optimal solution. Include the greedy solution and the optimal solution in your answer.

4. Identify two subsets $i, j \subset E$ such that $(E, \mathcal{I} + i + j)$ is a matroid.

**Question 3:** Give a linear-time algorithm that takes as input a tree $T$ and determines whether it has a perfect matching, i.e., whether there exists a subset of edges that touches each node exactly once.

**Question 4:** A subsequence is palindromic if it is the same whether read left to right or right to left. For instance, the sequence

$$A, C, G, T, G, T, C, A, A, A, A, T, C, G$$

has many palindromic subsequences, including A, C, G, C, A and A, A, A, A. Devise an algorithm that takes a sequence $x_1, x_2, \ldots, x_n$ and returns the length of the longest palindromic subsequence. For full marks the running time should be $O(n^2)$ but partial marks are available for less efficient solutions.

Hint: Let $A[i, k]$ be the length of the longest palindromic subsequence of $x_i, x_{i+1}, \ldots x_{i+k-1}$ and consider computing $A[1, n]$ via dynamic programming.

**Question 5:** Given a **sorted** list of **distinct integers** $A[1], A[2], \ldots, A[n]$, you want to find out whether there is an index $i$ for which $A[i] = i$. Give an algorithm that runs in time $O(\log n)$. Remember to prove that your algorithm is correct and analyze the running time.

For example, $A = [-1, 1, 3, 5, 6, 7]$ does have such an index, i.e., $A[3] = 3$. But $A = [0, 1, 2, 5, 6, 7]$ doesn't have such an index.