NAME: ____________________

CMPSCI 611
Advanced Algorithms
Midterm Exam Fall 2015

A. McGregor 21 October 2015

DIRECTIONS:

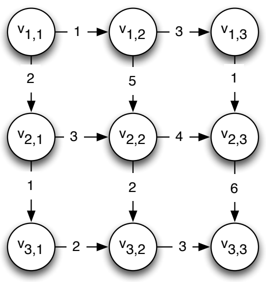
- Do not turn over the page until you are told to do so.
- This is a *closed book exam.* No communicating with other students, or looking at notes, or using electronic devices. You may ask the professor or TA to clarify the meaning of a question but do so in a way that causes minimal disruption.
- If 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 six 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 blank page 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. However, the best answers are those that are clear and concise (and of course correct).

1	/10
2	/10
3	/8+2
4	/10
5	/10
6	/10
Total	/58+2

Question 1 (The True or False Question). For each of the following statements, say whether they are TRUE or FALSE. No justification is required.

1. If a subset system satisfies the exchange property, it also satisfies the cardinality property.
2. Given a connected, undirected graph, let D be the diameter and let M be the size of the maximum matching. Then $M \geq D/2$.
3. Given a matroid $(E, \mathcal{I})$ there is at most one maximal solution $i \in \mathcal{I}$.
4. In a bipartite graph there are no cycles with an odd number of edges.
5. Given a undirected weighted graph G, let T be a minimum spanning tree. Then the length of the shortest path between u and v in G equals the length of the shortest path between u and v in T.

Question 2 (Grid Graphs). In this question, we develop an alternative shortest path algorithm for a specific type of graph. In a *grid graph* there are $n = k^2$ nodes labelled $v_{i,j}$ where i and j range between 1 and k. For each $i < k$ there is a "down" edge of length $d_{i,j}$ from $v_{i,j}$ to $v_{i+1,j}$. For each $j < k$ there is an "across" edge of length $a_{i,j}$ from $v_{i,j}$ to $v_{i,j+1}$. For example, a possible graph for $n = 9$ is



and in this graph, for example, the across edge from $v_{2,1}$ to $v_{2,2}$ has length $a_{2,1} = 3$.

1. What is the length of the shortest path from $v_{1,1}$ to $v_{3,3}$ in the above example?

2. Let $d[i, j]$ be the length of the shortest path from $v_{i,j}$ to $v_{k,k}$. If $i > k$ or $j > k$, let $d[i, j] = \infty$. Give a formula for $d[i, j]$ in terms of $d[i + 1, j]$ and $d[i, j + 1]$. Justify your answer.

3. Design an algorithm that computes $d[1, 1]$. Prove the running time and correctness.

4. Any path from $v_{1,1}$ to $v_{k,k}$ will use exactly $k-1$ down edges and $k-1$ across edges. Suppose we add the constraint that we only allow a path where the sub-path from $v_{1,1}$ to any intermediate node on the path uses at least as many across edges as down edges. How would you modify your algorithm to find the shortest such path?

Question 3 (Planning Your Working Vacation). Suppose you are a ski instructor and that you plan to work at a European ski resort for some part of the n-day ski season next year. Unfortunately, there isn't enough work for you to be paid every day and you need to cover your own expenses. Fortunately, you know in advance that if you are at the resort on the ith day of the season, you'll make p_i euros where p_i could be negative (if expenses are more than earnings) or positive (if expenses are less than earnings). To maximize your earning you should choose carefully which day you arrive and which day you leave; the days you work should be consecutive and you don't need to work all season. For example, if $n = 8$ and

$$p_1 = -9 \ , \ p_2 = 10 \ , \ p_3 = -8 \ , \ p_4 = 10 \ , \ p_5 = 5 \ , \ p_6 = -4 \ , \ p_7 = -2 \ , \ p_8 = 5$$

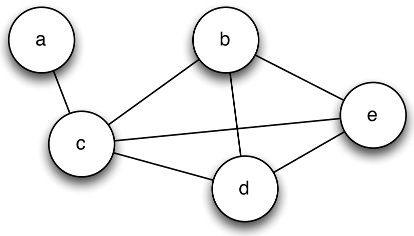
then if you worked from day 2 to day 5, you would earn $10 - 8 + 10 + 5 = 17$ euros in total.

1. Give a divide and conquer algorithm for this problem that runs in $O(n \log n)$ time. You may assume n is a power of two. If your algorithm takes longer or doesn't use divide and conquer you might get partial credit.

2. Justify why your algorithm is correct and analyze the running time.

3. **Extra Credit:** Design an $O(n)$ time algorithm for this problem.

Question 4 (Triangles and Four Cycles). Given an undirected graph G on n nodes, a triangle in the graph is a set of three different nodes $\{u, v, w\}$ such that there are edges between each of these nodes. For example, in the following graph $\{b, c, d\}$ is a triangle.



1. List the other triangles in the above graph.

2. Design an $O(n^\omega)$ time algorithm that returns the number of triangles in a graph where $O(n^\omega)$ is the running time of an algorithm that multiplies two $n \times n$ matrices. Recall $\omega < 2.38$. Analyze the running time and prove your algorithm is correct. **Hint:** If M is the adjacency matrix of G, first consider computing $M \times M$.

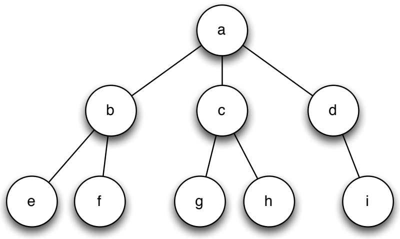
3. A four-cycle is a set of four different nodes $\{u, v, w, x\}$ such that there is an edge between u and v, v and w, w and x, and x and u (there may also be additional edges between the nodes). For example, in the example earlier, $\{b, c, d, e\}$ is a four-cycle. Design an $O(n^{\omega})$ time algorithm that returns the number of four-cycles in a graph.

Question 5 (Planning A Party). Alice wants to throw a party and is deciding whom to call. She has n (which is at least 11) people to choose from, and she has made up a list of which pairs of these people know each other. She wants to invite as many people as possible subject to the following two constraints:

1. Every person invited should know at least five other people that are invited.
2. Every person invited should not know at least five other people that are invited.

Design an efficient algorithm for maximizing the number of people she can invite. Remember to analyze the running time and correctness. **Hint:** Maximizing the number of invitees is the same as minimizing the number of people Alice doesn't invite. Obviously Alice might not be able to invite everyone. For example, if one of the n people knows less than five people out of the n potential invitees then the first constraint can never be satisfied for that person.

Question 6 (Independent Sets in Trees). Given a graph G, we say a set of nodes S is independent if there are no edges in G between any two nodes in S. In this question we consider the problem of computing independent sets when G is a tree. For example, in the following tree,



the set $S = \{b, g, h, d\}$ is an independent set. Suppose every node v has a weight $w(v)$. Design and analyze an algorithm for finding an independent set S in a tree such that $\sum_{v \in S} w(v)$ is maximized. Remember to analyze the running time and prove correctness. **Hint:** Consider a dynamic problem where there are n subproblems and each corresponds to finding the maximum weight independent set restricted to a node and its decedents. For example, in the subproblem for node c we are interested in finding the maximum weight independent subgraph amongst the nodes $\{c, g, h\}$.