THE AUSTRALIAN NATIONAL UNIVERSITY

Second Semester 2006

COMP3600/COMP6466
(Algorithms)

Writing Period: 2.5 hours duration

Study Period: 15 minutes duration

Permitted Materials: None

Answer ALL Questions

All your answers must be written in the boxes provided in this booklet. You may be provided with scrap paper for working, but it must **not** *be used to write final answers.*

There is additional space at the end of the booklet in case the boxes provided are insufficient. Label such overflow boxes with the question number.

Do not remove this booklet from the examination room.

Name (family name first):

Student Number:

Official use only:

Q1 (30)	Q2 (25)	Q3 (15)	Q4 (12)	Q5 (10)	Q6 (8)	Total (100)

QUESTION 1 [30 marks]

(a) Let $f(n)$, $g(n)$ and $h(n)$ be three positive functions. Prove or disprove the following conjectures.

(i) $\min\{f(n), g(n)\} = O(\max\{f(n), g(n)\})$.

QUESTION 1(a) (i)	**[5 marks]**

(ii) $f(n) = \Theta(g(n))$ implies $\log(f(n)) = \Omega(\log(\log g(n)))$,
where $\log(\log g(n)) > 1$ and $f(n) \geq 1$ for all sufficiently large n.

QUESTION 1(a) (ii).	**[5 marks]**

(b) Using the $O(.)$ notation, give asymptotic upper bounds for $T(n)$ in the following recurrences. Note that you are **not** allowed to use the Master theorem.

(i) $T(n) = 4T(n/4 + 7) + n/3$
(ii) $T(n) = T(\sqrt{n}) + 5n$

Assume that n is power of 2 and $T(n)$ is constant for $n \leq 4$.

QUESTION 1(b) (i) **[5 marks]**

QUESTION 1(b) (ii) **[5 marks]**

(c) Given n elements in a set S and k with $1 \leq k \leq n$, the following algorithm $\mathcal{A}$ similar to the given algorithm `SELECT` in the textbook can be used to find the kth largest element in S, which proceeds as follows.

Step 1. The elements in S are grouped with size 13 each, except the last group which contains $n - \lfloor n/13 \rfloor * 13$.

Step 2, Find the median of each group. As a result, there will be a sequence of medians consisting of the medians of the groups. Let x be the median of the median sequence.

Step 3. The set S is now partitioned into 3 disjoint sets R_1, R_2 and R_3 which contain elements less, equal to or greater than x.

Step 4. Call algorithm $\mathcal{A}$ for finding the $(|R_1| - (n-k+1)+1)$th largest element on R_1 recursively if $|R_1| \geq n-k+1$. Otherwise, if $|R_1 \cup R_2| \geq n-k+1$, return x; otherwise, call algorithm $\mathcal{A}$ for finding the kth largest element on set R_3 recursively.

Write a recurrence for the running time of algorithm $\mathcal{A}$ and show the algorithm works within the linear time.

QUESTION 1(c) (more room on next page) **[10 marks]**

Answer box continued over page.

QUESTION 1(c) continued. **[10 marks]**

QUESTION 2 [25 marks]

(a) Three probing techniques for hash tables with open addressing were introduced in the lectures. Describe the hash functions used by these techniques briefly. Design a new hash function for open address probing such that the number of different probing sequences derived from it is $O(m^3)$, where m is the size of the hash table.

QUESTION 2(a)	**[6 marks]**

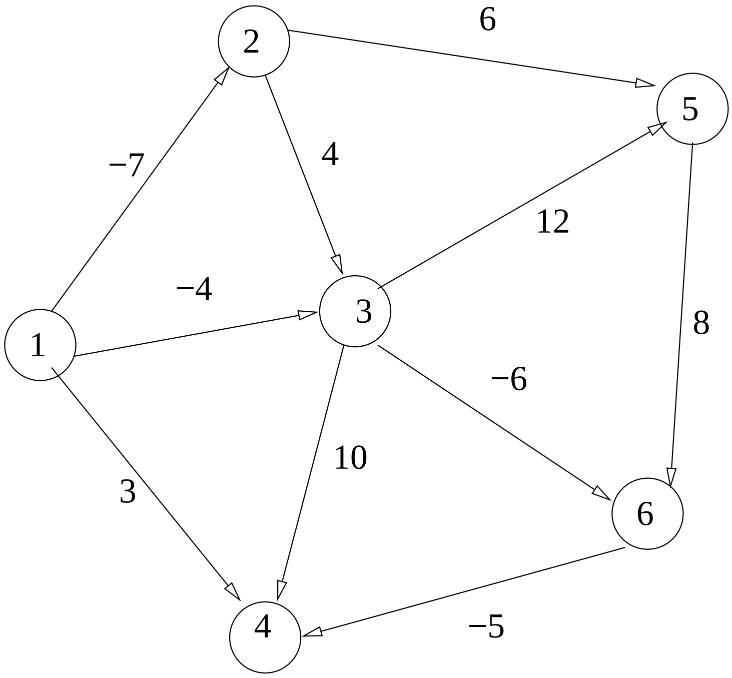
(b) Assume a given disjoint-set is represented by linked lists. To speed up the `UNION` operation on the sets, a heuristic has been introduced, what is it? Describe how to implement the `UNION` operation using the heuristic briefly.

QUESTION 2(b)	**[3 marks]**

(c) To maintain the properties of the red-black tree, a left rotation (or right rotation) operation is needed when dealing with keys insertion or deletion. Show the binary search tree property always holds when performing the **left rotation** operation in a red-black tree.

QUESTION 2(c)	**[4 marks]**

(d) (i) Perform a Depth-First Search on the directed acyclic graph (DAG) below, starting at vertex 1 and exploring neighbouring vertices **in order of their labels**. Give your results as a diagram that shows the order of vertex visiting and the start and finish times of each vertex. Label the edges in the graph as back edges, forward edges, cross edges, and tree edges if they exist.



QUESTION 2(d) (i)	**[6 marks]**

(ii) Find the shortest path from vertex 1 to every other reachable vertex in the DAG. What is the running time of your algorithm if the DAG contains m directed edges and n vertices?

QUESTION 2(d) (ii)	**[6 marks]**

QUESTION 3 [15 marks]

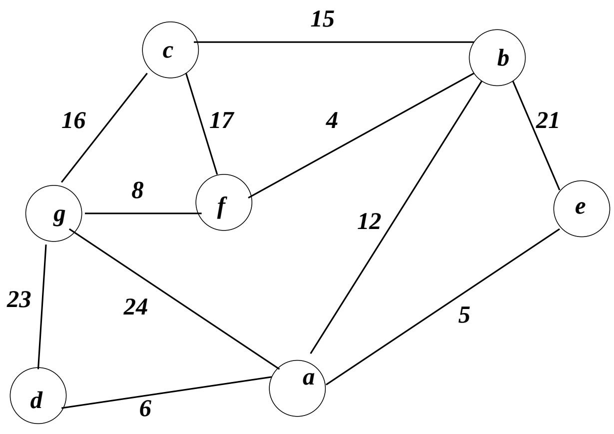
A telecommunications company is about to build a telecommunication network for a new city, let V be the node set and E the set of possible cable links. Associated with each possible link $e = (u, v) \in E$ that links two nodes u and v, there is a non-negative weight $w(e)$, which is the cost to lay a cable between them.

Now, the problem is to find an optimal strategy to make all the nodes in V be connected by laying cables among them such that the total cable cost is minimised.

(a) Which algorithm in the textbook can be used to find such an optimal strategy.

QUESTION 3(a)	**[5 marks]**

(b) Use the nominated algorithm on the following planning telecommunication network $G(V, E)$ to find an optimal strategy for it, and show each major step of the algorithm on G in a separate diagram.



QUESTION 3(b) (more room on next page)	**[10 marks]**

Answer box continued over page.

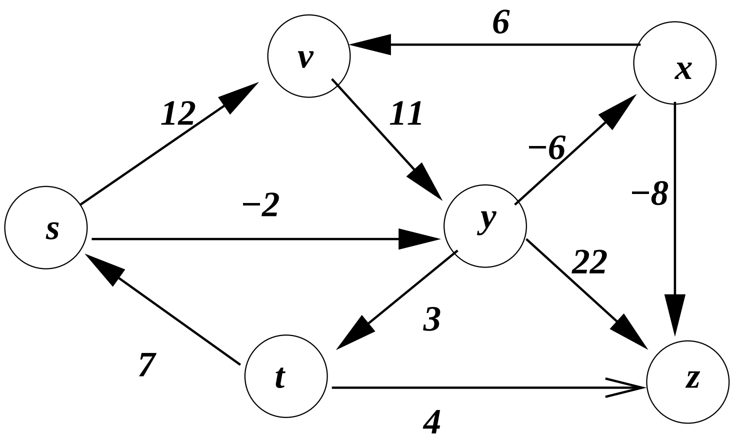
QUESTION 3(b) continued. **[10 marks]**

QUESTION 4 [12 marks]

(a) Explain why Dijkstra's algorithm is not applicable to find single-source shortest paths in a directed weighted graph that contains negative edge weights?

QUESTION 4(a)	**[3 marks]**

(b) Given the following directed graph $G(V, E)$ with source s, the problem is to find the shortest path from s to every other vertex $v \in V - \{s\}$.



Apply the Bellman-Ford algorithm to solve the problem, and show the **first three** main intermediate steps.

QUESTION 4(b) (more room on next page)	**[9 marks]**

Answer box continued over page.

QUESTION 4(b) continued. **[9 marks]**

QUESTION 5 [10 marks]

(a) Explain how Floyd-Warshall's algorithm can be used to determine whether a given directed graph is a DAG. Is it an efficient algorithm for this problem?

QUESTION 5(a)	**[5 marks]**

(b) How to apply Floyd-Warshall's algorithm to find the transitive closure of a directed graph? Describe the major steps of the proposed algorithm.

QUESTION 5(b)	**[5 marks]**

Answer box continued over page.

QUESTION 5(b) **[5 marks]**

QUESTION 6 [8 marks]

Suppose we are given an n-node rooted tree T with each node v in T being assigned a positive weight $w(v)$. An *independent set* S of T is such a subset of the nodes in T that no node in S is a child or parent of any other node in it. Devise an efficient **dynamic programming** algorithm to find the maximum-weighted independent set of T, where the weight of a set of nodes is simply the sum of the weights of the nodes in that set. What is the running time of your algorithm?

QUESTION 6 (more room on next page) **[8 marks]**

Answer box continued over page.

QUESTION 6 continued. **[8 marks]**

Additional answers. Clearly indicate the corresponding question and part.

Additional answers. Clearly indicate the corresponding question and part.

Additional answers. Clearly indicate the corresponding question and part.