Student Name:

Student Number:

<table>
<thead>
<tr>
<th>Question</th>
<th>Maximum Number of Marks</th>
<th>Marks Received</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>110</td>
<td></td>
</tr>
</tbody>
</table>

Instructions

1. **All questions must be answered on this examination paper.** You may use both sides of the answer sheets for answers. There are some extra sheets attached at the end of this examination paper.

2. If you find a question to be ambiguous or unclear, then make, and state, whatever assumptions you feel are necessary; your mark will then be partly based on the reasonableness of your assumptions.

3. Please answer to the point and do not write everything you know on the topic. Substantial marks will be deducted if your answer is not precise.

4. All logarithms are base 2.

5. Although the sum-total of all the marks is 110, the exam will be treated as if it is out of 100. Alternatively, you may think of that you need not attempt each and every question.
Question 1: (2*15=30 marks) Answer TRUE or FALSE. Correct answer is worth (+2) Marks. Wrong answer is worth (-1) Marks. No answer is worth (0) Marks.

1. Is $\sqrt{1} + \sqrt{2} + \sqrt{3} + \ldots + \sqrt{n} = O(n^{3/2})$?

2. If the SUBSET SUM problem can be solved in polynomial time then $P=NP$?

3. Is $P \subseteq NP$?

4. Let $L_1$ and $L_2$ be two languages corresponding to two decision problems and let $L_1 \leq_P L_2$ (under the polynomial time reducibility, where a binary string $x$ is mapped to the binary string $f(x)$ in polynomial time). Does this imply that if the membership of $x$ in $L_1$ can be tested in polynomial time then the membership of $f(x)$ in $L_2$ can also be tested in polynomial time?

5. Dynamic Programming is a technique to solve NP-Complete Problems in Polynomial Time.

6. A min-Heap on $n$-elements can be constructed in $O(n)$ time.

7. Given a min-Heap on $n$-elements, the elements can be reported in sorted order in $O(n)$ time.

8. There are two algorithms for a problem. The running time of Algorithm I is expressed as $T(n) = T(n/2) + T(n/4) + n$ and the running time of Algorithm II is expressed as $T(n) = T(n/3) + T(2n/3) + n$. Is the running time of Algorithm II smaller than that of Algorithm I for large values of $n$?
9. If a graph $G$ on 8 vertices has a clique of size 5, does the complement of $G$ has a vertex cover of size 3?

10. Given a graph $G = (V, E)$, it can be decided within $O(|V| + |E|)$ time whether there is a path between two nodes in $V$ consisting of at most 10 edges.

11. It is sufficient to perform 70 multiplications in order to multiply four matrices $A \times B \times C \times D$, where dimensions of matrix $A$ is $3 \times 5$, $B$s is $5 \times 6$, $C$s is $6 \times 1$ and $D$s is $1 \times 2$.

12. It takes $O(n)$ time to find the 100th smallest number in a set consisting of $n$ real numbers.

13. In the worst case the height of a Red-Black Binary Search Tree on $n$ nodes can be $\Omega(n)$.

14. Your task is to sort the first names of all the students in COMP 3804 class, where each name is at most 10 character long and only uses the characters from English alphabets. Assume that the total number of students in the class is $n$. Can these names be sorted in lexicographic order (i.e. dictionary order) in $O(n)$ time?

15. The recurrence

$$T(n) \leq \begin{cases} T(\lceil n/5 \rceil) + T(\lceil 7n/10 \rceil) + O(n) & \text{if } n > 140, \\ O(1) & \text{if } n \leq 140. \end{cases}$$

evaluates to $T(n) = O(n)$. 

Question 2: (5 marks) Let $A$ and $B$ be two sets of real numbers, each set containing $n$ elements. Describe an algorithm (in plain English) that returns `true` if there is an element $a \in A$ and an element $b \in B$ such that $|a - b| \leq 1$. If such elements $a$ and $b$ do not exist, then the algorithm returns `false`. The running time of your algorithm must be $O(n \log n)$. Explain why your algorithm is correct and why its running time is $O(n \log n)$. 
Question 3: (10 marks) Show why any comparison sort algorithm requires $\Omega(n \log n)$ comparisons in the worst case. The proof need not be formal - but the main ideas need to be made clear.
Question 4: (5+5+5+5 marks)

(4.a) State clearly the steps in Dijkstra’s Single Source Shortest Path Algorithm for a directed weighted (positive weights) graph $G = (V, E)$. (Just the steps - as Steps 1, 2, 3, ..)

(4.b) Analyze the complexity of Dijkstra’s algorithm, by analyzing each of the steps of the algorithm in Part a.
(4.c) In the algorithm we maintain \( d(v) \) (i.e., the tentative distance) for each vertex \( v \). For each vertex \( v \), other than the source vertex \( s \), \( d(v) \) is initialized to \(+\infty\), and during the algorithm \( d(v) \) is updated and it reflects the least cost path from \( s \) to \( v \) which the algorithm has found so far. Prove that \( d(v) \geq \delta(s, v) \), for all \( v \in V \), at any stage of the algorithm, where \( \delta(s, v) \) denote the cost of a shortest path between \( s \) to \( v \).

(4.d) Illustrate for the graph below how your algorithm from Part (a) computes the lengths of shortest paths from the vertex \( s \) to each of the four other vertices.
Question 5: (5+5+5=15 marks)
Which of the following algorithms result in a minimum spanning tree? Justify your answer. Assume that the graph $G = (V, E)$ is a simple, undirected, weighted and connected.

1. Sort the edges with respect to decreasing weight.
   Set $T := E$.
   For each edge $e$ taken in the order of decreasing weight do, if $T - \{e\}$ is connected, then discard $e$ from $T$.
   Set $MST(G) = T$.

2. Set $T := \emptyset$.
   For each edge $e$, taken in arbitrary order do, if $T \cup \{e\}$ has no cycles then $T := T \cup \{e\}$.
   Set $MST(G) = T$.

3. Set $T := \emptyset$.
   For each edge $e$, taken in arbitrary order do
   
   begin
   $T := T \cup \{e\}$.
   If $T$ has a cycle $c$ then let $e'$ be a maximum weight edge on $c$.
   Set $T := T - \{e'\}$.
   end
   Set $MST(G) = T$. 

Question 6: (15 marks) Using dynamic programming provide an algorithm running in $O(Wn)$ time for the following problem: we are given a set $X$ of $n$-items, where weight of the $i$-th item is $w_i$ kilograms and its dollar value is $v_i$ ($1 \leq i \leq n$). We need to fill in a bag, which can hold a weight of at most $W$ kilograms, by items from $X$, so that the sum total of the dollar values of all the items placed in the bag is maximized. All the quantities $w_i$'s, $v_i$'s and $W$ are positive integers.

(Hint: Let $X = \{x_1, x_2, \ldots, x_n\}$ be the set of $n$-items. Let $k(w, j)$ denote the subproblem of finding the maximum dollar value that can be achieved for a bag of capacity $w$ kilograms, where items are chosen from the set $\{x_1, \ldots, x_j\}$. Observe that we are interested in solving $k(W, n)$. Express the subproblem $k(w, j)$ in terms of smaller subproblems and use dynamic programming.)
Question 7: (5 marks)
What are the main elements of a greedy algorithm? Justify this using an example.
Question 8: (5 marks) Show that the SET-PARTITION problem is NP-Complete. In the set partitioning problem, the input is a set $P$ of numbers. The decision problem is whether the numbers can be partitioned into two sets $A$ and $\bar{A} = P - A$, such that $\sum_{x \in A} x = \sum_{x \in \bar{A}} x$. (Hint: Recall the SUBSET-SUM problem, where you are given a set $S$ of numbers and a target value $t$, and the decision problem was whether there is a subset $S' \subseteq S$, such that $\sum_{x \in S'} x = t$. Show that SUBSET-SUM $\leq_P$ SET-PARTITION, i.e., an instance $< S, t >$ of SUBSET-SUM can be reduced to an instance $< P >$ of SET-PARTITION. Set $P$ can be constructed using the elements in set $S$ and an additional number $q$, where $t + q = \frac{1}{2}(q + \sum_{x \in S} x)$.)
Question 9: (5 marks) An independent set of a graph $G = (V, E)$ is a subset of vertices $V' \subseteq V$ such that each edge in $E$ is incident on at most one vertex in $V'$. The independent set problem is to find a maximum size independent set in $G$. Formulate a related decision problem for the independent set problem and prove that it is NP-Complete.