| Candidate Number |
| --- |
| |

**G6017**

# THE UNIVERSITY OF SUSSEX

**BSc and MComp SECOND YEAR EXAMINATION**
**January 2021 (A1)**

## Program Analysis

**Assessment Period: January 2021 (A1)**

# DO NOT TURN OVER UNTIL INSTRUCTED TO BY THE LEAD INVIGILATOR

**Candidates should answer TWO questions out of THREE.**
**If all three questions are attempted only the first two answers will be marked.**

***The time allowed is TWO hours.***

***Each question is worth 50 marks.***

**At the end of the examination the question paper and/or answer book, used or unused, will be collected from you before you leave the examination room.**

1.

(a) Precisely specify the conditions under which the following algorithm returns True, and then discuss, in detail, the running time of the algorithm. If you think it has different best- and worst-case running times then these should be considered separately, and you should explain the conditions under which best and worst-cases arise.

You must fully explain your answer and use O, Ω and θ appropriately to receive full marks.

---

Algorithm Ex1 $\big((a_1, \ldots a_n), (b_1, \ldots, b_m)\big)$

```
k ← n × m
for i ← 1 to n do
        for j ← 1 to m do
                If a_i < b_j
                        k ← k − 1

return k > 0
```

---

[10 marks]

(b) Precisely specify the conditions under which the following algorithm returns True, and then discuss, in detail, the running time of the algorithm. If you think it has different best- and worst-case running times then these should be considered separately, and you should explain the conditions under which best and worst-cases arise.

You must fully explain your answer and use O, Ω and θ appropriately to receive full marks.

---

Algorithm Ex2 $\big((a_1, \ldots a_n), (b_1, \ldots, b_n)\big)$

```
q ← true
for i ← 1 to n do
        j ← n
        while j > 0 and q == true do
                If a_i ! = b_j
                        q ← false
                j ← j − 1

return q
```

---

[10 marks]

(c) Precisely specify the conditions under which the following algorithm returns True, and then discuss, in detail, the running time of the algorithm. If you think it has different best- and worst-case running times then these should be considered separately, and you should explain the conditions under which best and worst-cases arise.

You must fully explain your answer and use O, Ω and θ appropriately to receive full marks.

---

Algorithm Ex3 $(T, c)$

$T$ is a balanced binary search tree containing integer number values and $c$ is an integer. Each node in $T$ has two child nodes, either a numeric value or the value $null$ to indicate the lack of a child value.

$node \leftarrow root(T)$
$found \leftarrow false$

while $(node\ != null)\ and\ (found == false)$
    if $node.value == c$
        $found \leftarrow true$
    else if $c < node.value$
        $node = node.leftChildNode$
    else
        $node = node.rightChildNode$

return $found$

---

[10 marks]

(d) A data pattern analyser is to be built that can detect and count up the number of occurrences of two and three letter sub sequences in a sequence (e.g. (x,y) or (c,a,t) ). The analyser should stop if it encounters ***** in the sequence and return the number of occurrences found up to that point. So, for example, consider that we are searching for:

- The two letter sequence (a,b)
- The three letter sequence (b,a,d)

| **Input string** | **2 letter sub sequences found** | **3 letter sub sequences found** |
|---|---|---|
| (a,b,a,b,b,a,b,*****) | 3 | 0 |
| (a,b,a,d,a,b) | 2 | 1 |
| (b,a,b,a,d,*****) | 1 | 1 |
| (*****,a,b) | 0 | 0 |
| (a,b,a,d,*****,a,b,a,d) | 1 | 1 |

Produce a formal statement of this problem, and then write an algorithm to solve the problem using a pseudo code style similar to the one shown in parts (a) to (c). State the bounds on the best and worst case performance of your algorithms using O, $\Omega$ and $\theta$ appropriately to receive full marks.

[10 marks]

(e) The Stable Match problem is concerned with matching $n$ members of one set $A = \{a_1, \ldots, a_n\}$ with $n$ members of a set $B = \{b_1, \ldots, b_n\}$, according to a set of preferences. The aim is to produce a stable and perfect match. An algorithm to solve this problem is presented below (the Gale and Shapley algorithm we studied in our class).

```
Initialise each element in A and B to be free
While an element from A is free and has elements in B still to propose to
        Choose an element a from A
        Let b be the next best option b ∈ B from a's preference list
        If b is free
                Match a and b
        Else if b prefers a to b's existing match denoted b_current
                Match a and b and set b_current to be free
        Else
                b rejects a
```

Explain what the complexity classes are for the best and worst case running time for this algorithm. You answer should include a diagram to illustrate one example each of the conditions where the best and worst case would occur. Each diagram should show the sets $A$ and $B$, their preferences and the final match pairings produced by the algorithm. Use a size for sets $A$ and $B$ of 3 in your diagrams.

[10 marks]

2.

(a) A student has been asked to put some parcels on a shelf. The parcels all weigh different amounts, and the shelf has a maximum safe loading weight capacity of 150 Kg. The weight of parcels are as follows (in Kg):

| $\boldsymbol{parcel}$ | $\boldsymbol{weight\ (Kg)}$ |
|---|---|
| 1 | 60 |
| 2 | 10 |
| 3 | 50 |
| 4 | 25 |
| 5 | 10 |
| 6 | 20 |
| 7 | 10 |

The student has been asked to load the maximum weight possible parcels on the shelf subject to the maximum safe loading weight.

State two possible approaches for a greedy algorithm solution to solve this problem. In each case, state clearly the result you would get from applying that approach to this problem, stating whether the solution is optimal or not. If your answer does not produce an optimal solution, what algorithm could be employed to find one?

[10 marks]

(b) Explain, using diagrams as appropriate, why Dijkstra's algorithm for finding the lowest cost path through a weighted graph is an example of a greedy algorithm. Your answer should include the key concept of an invariant.

[5 marks]

(c) A balanced binary heap is to be constructed using the following data elements:

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 95 | 56 | 34 | 86 | 41 | 17 | 29 | 32 | 58 | 3 | 4 | 11 | 60 | 77 | 82 |

using a bottom up heap construction method.

(i) Explain how the bottom up construction method can be shown to have running time complexity $O(n)$ where $n$ is the number of elements in the binary heap.

You may find it helpful to recall that:

$$\sum_{h=0}^{h=\infty} \frac{h}{2^h} \to 2$$

I.e. that $\frac{0}{2^0} + \frac{1}{2^1} + \frac{2}{2^2} + \frac{3}{2^3} + \cdots + \frac{h}{2^h} \to 2\ as\ h \to \infty$

[10 marks]

(ii) Show the heap that results from the bottom up construction. Fill the heap levels starting from the lowest first, and working from the right hand side to the left hand side.

[5 marks]

(d) A recursive algorithm is applied to some data $A = (a_1, \dots, a_m)$ where $m \geq 2$ . The running time T is characterised using the following recurrence equations:

$T(2) = c$ when the size of $A$ is 2
$T(m) = T(m-1) + 2c$ otherwise

Determine the running time complexity of this algorithm.

[10 marks]

(e) Another recursive algorithm is applied to some data $A = (a_1, \dots, a_m)$ where $m = 2^x$ (i.e. 2, 4, 8,16 …) where $x$ is an integer $\geq 1$. The running time T is characterised using the following recurrence equations:

$T(1) = c$ when the size of $A$ is 1
$T(m) = 2T\left(\frac{m}{2}\right) + c$ otherwise

Determine the running time complexity of this algorithm.

[10 marks]

3.

(a) The subset sum problem can be reliably solved optimally using the dynamic programming algorithm shown below:

```
SubsetSum(n, W)

Let B(0, w) = 0 for each w ∈ {0, ..., W}
  for i ← 1 to n
    for w ← 0 to W
      if w < w_i then
        B(i, w) ← B(i − 1, w)
      else
        B(i, w) ← max (w_i + B(i − 1, w − w_i), B(i − 1, w))
```

Where $n$ is the number of requests, $W$ is the maximum weight constraint, $w_i$ is the weight associated with request $i$, and $B$ in the solution space. You are given a set of requests and their corresponding weights. The maximum weight constraint $W$ is 12.

| $i$ | $w_i$ |
|---|---|
| 1 | 1 |
| 2 | 5 |
| 3 | 1 |
| 4 | 2 |
| 5 | 1 |
| 6 | 10 |

Copy the following solution space table to your answer book (do not write your answer on the question paper) and complete the table to determine the optimal subset sum.

| | | $w$ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| $i$ | **6** | | | | | | | | | | | | | |
| | **5** | | | | | | | | | | | | | |
| | **4** | | | | | | | | | | | | | |
| | **3** | | | | | | | | | | | | | |
| | **2** | | | | | | | | | | | | | |
| | **1** | | | | | | | | | | | | | |
| | **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

[15 marks]

(b) Huffman coding is an efficient means of coding a message so that it occupies the smallest number of average bits/symbol.

A message of length $x$ symbols is made from a set of 8 symbols drawn from a vocabulary shown below, with each symbol being coded using 3 binary bits:

| **Symbol element** | **3-bit coding** | | | **Relative frequency** |
|---|---|---|---|---|
| A | 0 | 0 | 0 | 0.18 |
| B | 0 | 0 | 1 | 0.1 |
| C | 0 | 1 | 0 | 0.05 |
| D | 0 | 1 | 1 | 0.05 |
| E | 1 | 0 | 0 | 0.4 |
| F | 1 | 0 | 1 | 0.06 |
| G | 1 | 1 | 0 | 0.04 |
| H | 1 | 1 | 1 | 0.12 |

So the message HEDGE (x = 5) would be coded as 111,100,011,110,100. This message requires a constant 3 bits/symbol. Analysis shows that in general usage, the symbols appear with the relative frequency as shown also in the table.

(i) Construct a Huffman coding tree to determine the most efficient alternative coding for the symbols so as to minimise the number of bits/symbol to code a message.

[10 marks]

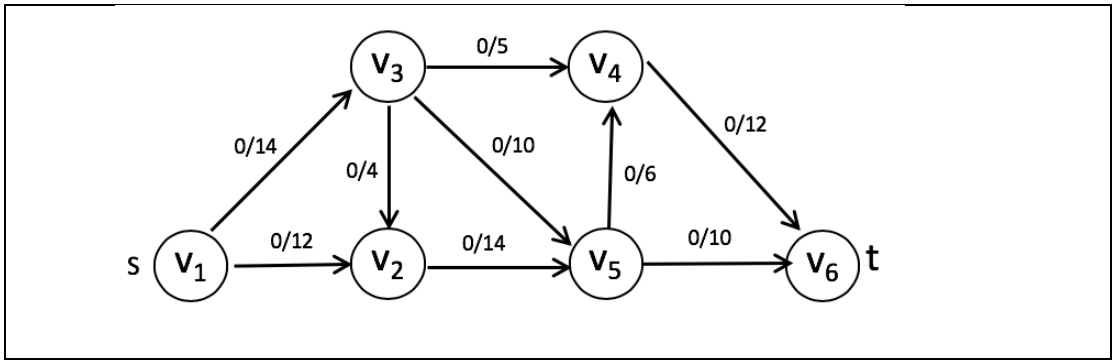
(ii) Determine the average number of bits/symbol required to code a message using this more efficient alternative coding.

[5 marks]

(c) The Ford-Fulkerson algorithm is used to determine network flow. The diagram below represents a data network that connects a Data Service Provider (DSP) connected to $v_1(s)$ to a customer connected to $v_6(t)$. Each edge represents a single data transmission link.

The notation $p/q$ indicates a current actual forwards flow $p$ measured in Gb/s in a pipe with a maximum capacity of $q$ also measured in Gb/s.

At the outset no data is being sent by the DSP to the customer.




i. Show the residual graph that will be created from the initial empty flow. When drawing the residual graph, show a forward edge with capacity $x$ and a backward edge with flow $y$ by annotating the edge $\vec{x}; \vec{y}$ .

[2 marks]

ii. What is the bottleneck edge of the path $(s, v_3, v_4, t)$ in the residual graph you have given in answer to part (a) ?

[2 marks]

iii. Show the residual graph after incorporating the simple path $(s, v_3, v_4, t)$ that results from augmenting the flow based on the residual graph you have given in answer to part (a).

[4 marks]

iv. Repeat the process outlined above incorporating additionally the simple paths $(s, v_3, v_2, v_5, t)$ , $(s, v_2, v_5, t)$ , $(s, v_3, v_5, v_4, t)$ and $(s, v_2, v_5, v_4, t)$ , showing each residual graph, to determine the maximum flow between $s$ and $t$, and thus the maximum data bandwidth that can be achieved between the DSP and the customer.

[12 marks]

**End of paper**