University of Sussex

Program Analysis G6017

Coursework 1

Due:	Semester 1 Week 7 – Wednesday 7 November 2018 by 4PM
Format:	Paper submission via the Engineering and Informatics School Office
Weighting	50% of the coursework element for this module

General instructions

1. Answer all of the questions.
2. Show your workings where appropriate. You can still get credit for a question with an incorrect final answer if your workings show that you understood what the problem was and how to solve it.
3. Do not copy the work of another student. Plagiarism is a very serious matter. Discussion between students is to be encouraged – copying is an academic disciplinary matter.
4. Check that you provide any working or information that the question asks for.
5. Hand your submission in on time. There are penalties for late submission.
6. If I cannot read your submission, I cannot mark it. It is your responsibility to ensure that the presentation of your submission is appropriate for a University student.
7. Do not forget to state units if they are relevant and apply to a question.
8. You should use any calculating aids your feel appropriate to help you solve the problems including, although not limited to, calculators, spreadsheets such as Excel and MATLAB.
9. If you do not understand the questions, you can get help at the workshop sessions.
10. This assignment is marked out of a total of 100.

../

Q1) Consider the following six functions. They are the running times of six algorithms. Arrange them in ascending order of asymptotic running time, and give an explanation of your ordering.

$$f_1(n) = 12n^3 + n^2\log(n)$$
$$f_2(n) = 48\sqrt{(n+45)} + \log(n)$$
$$f_3(n) = 3\ log(n^2)$$
$$f_4(n) = 3^n + 100n$$
$$f_5(n) = 2^n\log(n)$$
$$f_6(n) = 150000n^2$$

[15 marks]

Q2)

Specify the running time of each of the following algorithms. You must fully explain your answer to receive full marks.

(a)

```
Algorithm Ex1 $((a_1, \dots, a_n), b)$:

    $x \leftarrow 0$
    for $i \leftarrow 1$ to $n$ do
        if $a_i > b$
        $x \leftarrow x + a_i$
    return $x$
```

[7 marks]

(b)

```
Algorithm Ex2 $((a_1, \dots, a_n), (b_1, \dots, b_m))$:

    $x \leftarrow 0$
    for $i \leftarrow 1$ to $\min(n, m)$ do
        if $a_i > b_i$
        $x \leftarrow x + b_i$
    return $x$
```

[7 marks]

(c)

```
Algorithm Ex3 $\big((a_1, \dots, a_n), (b_1, \dots, b_n), (c_1, \dots, c_n)\big)$:

    $x \leftarrow 0$
    for $i \leftarrow 1$ to $n$ do
        for $j \leftarrow 1$ to $n$ do
            for $k \leftarrow 1$ to $n$ do
                $x \leftarrow x + (a_i \times b_j \times {c_k}^2)$
    return $x$
```

[7 marks]

(d)

```
Algorithm Ex4 $\big((a_1, \dots, a_n), (b_1, \dots, b_m)\big)$:

    $x \leftarrow 0$
    $i \leftarrow 1$
    while $i < n$ and $a_i > 0$ do
        for $j \leftarrow i$ to $m$ do
            $x \leftarrow x + a_i \times b_j$
        $i \leftarrow i + 1$
    return $x$
```

[7 marks]

The last part is concerned with the concept of proof by contradiction. Providing a clear statement of your steps, show using proof by contradiction that:

(e) The sum of an even and odd number is always odd.

[7 marks]

Q3) You are helping to design an encryption algorithm. The decryption phase of the algorithm takes two inputs, a message of length a digits and a key made up of n binary bits. The decryption phase performs a mathematical process on the message using the key to produce the plain text decoded message. You are very aware that there are hackers that would wish to be able to decode your messages. However, the workings of the algorithm have been kept secret so the only hack available for decoding the message is a brute force approach where each possible key is tried one at a time until the correct one is found, when a decoded message in English is produced. There are no known cribs, weaknesses or other techniques to assist in hacking the cipher.

(a) What is the running time for a brute force method to decode a message?

[5 marks]

(b) Should the lower bound you identified in part (a) concern us in terms of the safety and security of our encryption technique?

[5 marks]

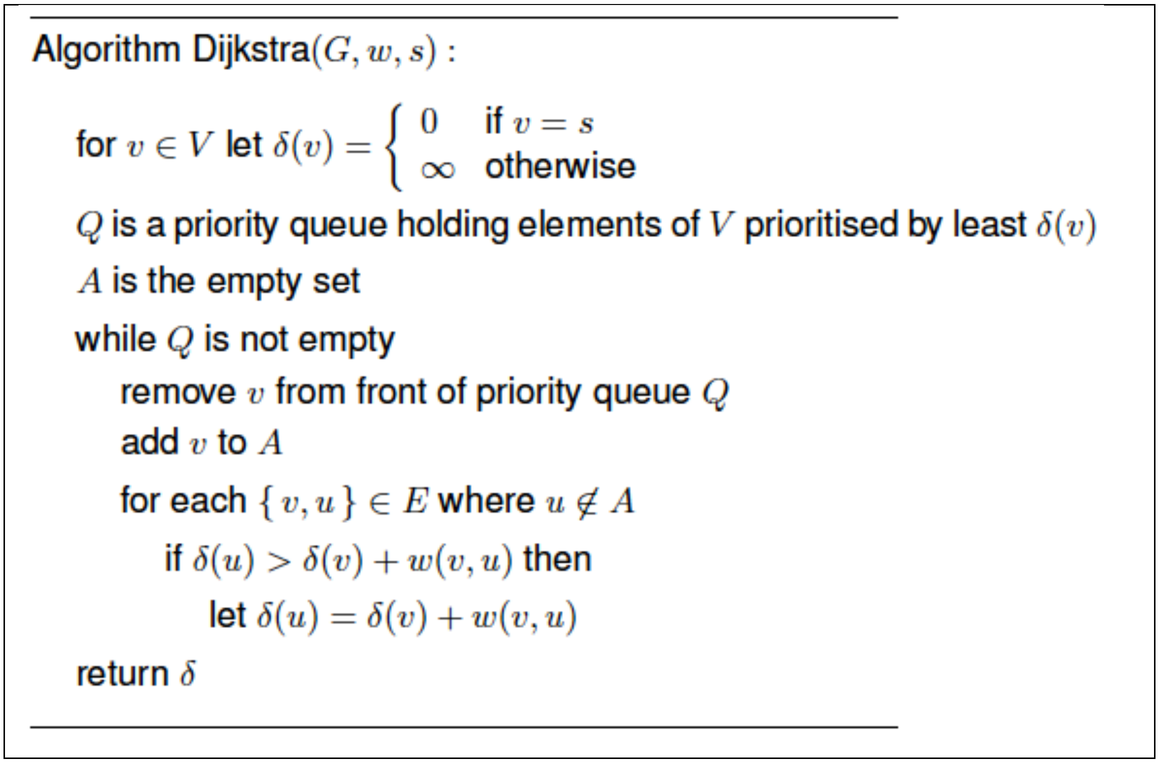
(c) The security of the encryption process depends on the value n. The decryption process is quite complex. Assume it takes 1 minute of processing time on a PC to apply a key (regardless of the value of n) to the message (regardless of whether it's the correct key). The algorithm may be regarded as sufficiently secure provided that, using a brute force method, there is no more than a 1% chance that the message would be decoded correctly in 30 days (working 24 hours a day). Calculate the minimum number of bits n that the key must be composed of.

[10 marks]

(d) In reality, the time it takes to apply the key depends on the value of n. If the key is short, the application time is quick. If the key is long, it takes longer to apply. Assume that the time to apply the key to the message is given by the formula $t = 0.5 \times n$ where n is the number of bits in the key and t is the time in seconds. This time the algorithm may be regarded as sufficiently secure provided that, using a brute force method, there is no more than a 0.5% chance that the message would be decoded correctly in 30 days (working 24 hours a day). Calculate the minimum number of bits n that the key must be composed of.

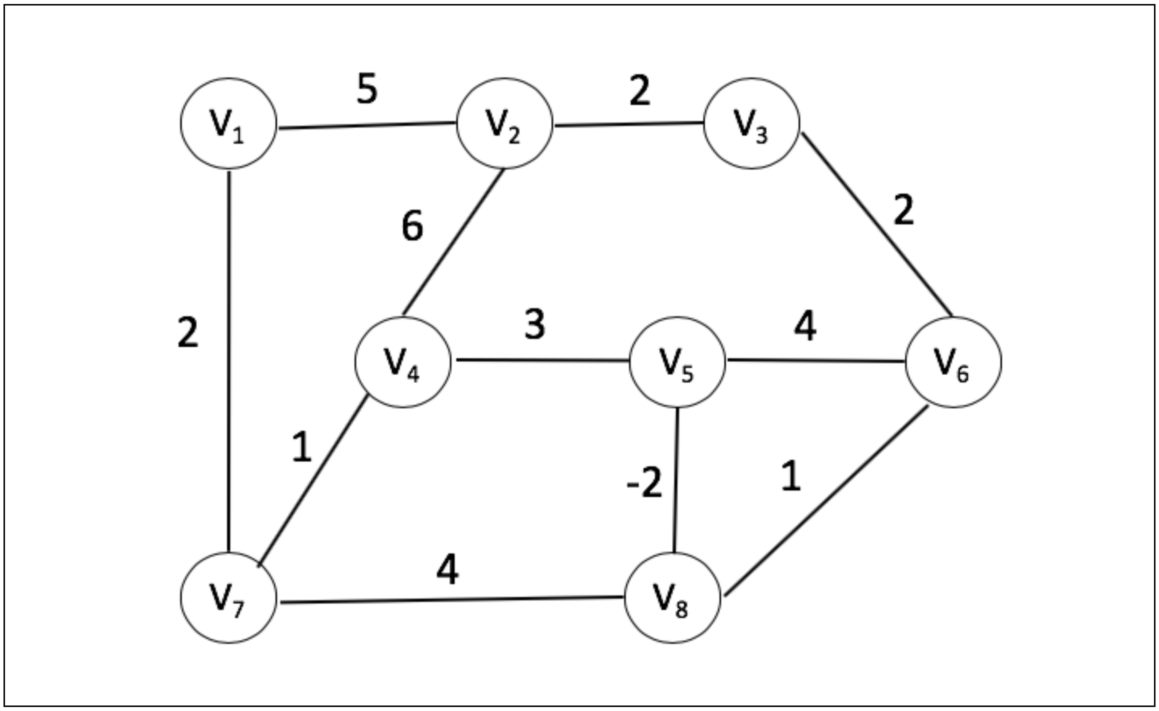
[10 marks]

Q4) This question concerns Dijkstra's algorithm which is reproduced overleaf:



```
Algorithm Dijkstra$(G, w, s)$ :
    for $v \in V$ let $\delta(v) = \begin{cases} 0 & \text{if } v = s \\ \infty & \text{otherwise} \end{cases}$
    $Q$ is a priority queue holding elements of $V$ prioritised by least $\delta(v)$
    $A$ is the empty set
    while $Q$ is not empty
        remove $v$ from front of priority queue $Q$
        add $v$ to $A$
        for each $\{v, u\} \in E$ where $u \notin A$
            if $\delta(u) > \delta(v) + w(v, u)$ then
                let $\delta(u) = \delta(v) + w(v, u)$
    return $\delta$
```

You will be considering what happens when the algorithm is run on the following graph where the source vertex is v_1. Note that this graph contains a negatively weighted edge.



(a) Complete the following table:

- The top row shows the values of δ after initialisation.
- In each of the subsequent rows, you should give the values of δ that all 8 vertices have at the end of that iteration of the algorithm's while loop.
- In the row for iteration i you should circle the value $\delta(v)$ (which you will have put in the column for the vertex v) for the vertex v that is selected in the first line of the while loop during the *i*-th iteration.
- Also, in the row for iteration i, you should show in bold (or with an underline) any value of δ that has changed on that iteration.

Then state whether you think that the answer is correct.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
iteration	0	∞	∞	∞	∞	∞	∞	∞
1								
2								
3								
4								
5								
6								
7								
8								

[10 marks]

(b) Give a concise description of the situations where giving Dijkstra's algorithm a graph with one or more negatively weighted edges will result in invalid output. You may find it helpful to use a diagram to assist in your description.

[10 marks]

Dr Kingsley Sage
Khs20@sussex.ac.uk
October 2018

END.