University of Sussex

Program Analysis G6017

Coursework 2

Due:	Semester 1 Week 12 – Wednesday 12 December 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)

This question is concerned with the design and analysis of recursive algorithms.

You are given a problem statement as shown below. This problem is concerned with finding the maximum and minimum values in a sequence A of real numbers. Whilst this could be done using a conventional loop based approach, your answer must be developed using a recursive algorithm. No marks will be given if your answer uses loops.

$FindMaxAndMin(a_1, \dots, a_n)$

Input: A sequence of real values $A = (a_1, \dots, a_n)$
Output: A pair (max, min) containing the maximum and minimum values in A.

Your recursive algorithm should use a single recursive structure to find the maximum and minimum values, and should not use two separate instances of a recursive design.

(a) Produce a pseudo code design for a recursive algorithm to solve this problem.

[5 marks]

(b) Draw a call-stack diagram to show the application of your recursive algorithm when called using the sequence $A = (5,3,1,7,2,6)$.

[5 marks]

(c) Write down the set of recurrence equations for your recursive algorithm. The recurrence equations should take a form like this:

$$T(n) = \begin{cases} (expression)\ if\ (expression) \\ (expression)\ otherwise \end{cases}$$

The “otherwise” case here refers to the recursive algorithm base case.

[4 marks]

(d) Using the recurrence equations you gave in your answer for part (c), determine the running time complexity of your recursive algorithm,

[6 marks]

Q2)

A piece of code implementing a recursive algorithm has been produced, and a student has analysed the recurrences. They have produced the recurrence equations as shown below:

$$T(n) = T(n-1) + \frac{n-1}{2} + c_1$$

$$T(2) = c_2$$

So the recursive algorithm features a base case when the size of the problem is $n = 2$. The values of c_1 and c_2 are constants.

(a) Determine the running time complexity of this recursive algorithm. To get the full marks, your analysis should be as complete as possible. To get an idea of how to perform a complete analysis, refer to the example recursive algorithm analysis on Canvas.

You will find it useful to know that the formula for a sum of an arithmetic sequence of numbers of the form $(1,2,3, \ldots . k)$ is given by the formula:

$$\sum_{n=1}^{n=k} n = \frac{k(k+1)}{2}$$

[20 marks]

Q3)

This question is concerned with dynamic programming.

A bottom up dynamic programming method is to be used to solve the subset sum problem. The problem is to find the optimal sum of weighted requests from a set of requests A subject to a weight constraint W.

The set of weighted requests $A = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}$ can be summarised as following:

Request	$\boldsymbol{w(\alpha_i)}$
α_1	2
α_2	6
α_3	3
α_4	4
α_5	11

The maximum weight constraint is 12.

Using the following algorithm (reproduced from the notes on Canvas):

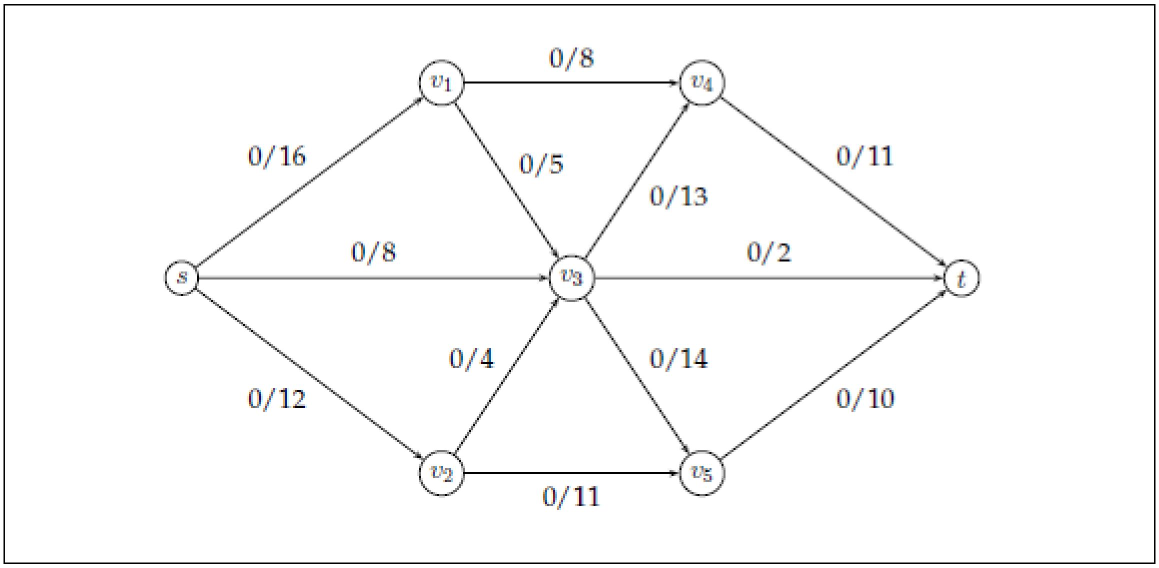
```
SubsetSum(n, W):
Let B(0, w) = 0 for each w ∈ {0, ..., W}
for i ← 1 to n
    for w ← 0 to W
        if w < w_i then
            B(i, w) ← B(i − 1, w)
        else
            B(i, w) ← max (w_i + B(i − 1, w − w_i), B(i − 1, w))
```

(a) Produce a table showing the space of the problem and all of the sub problems, and use that table to determine the optimal subset sum of requests when the weight constraint of 12 is applied. The table should take the form of a matrix with 6 rows (values of i in the range 0 to 5 inclusive) and 13 columns (values of w in the range 0 to 12 inclusive).

[20 marks]

Q4)

In this question, we consider the operation of the Ford-Fulkerson algorithm on the network shown below:



Each edge is annotated with the current flow (initially zero) and the edge's capacity. In general, a flow of x along an edge with capacity y is shown as x/y.

(a) Show the residual graph that will be created from this network with the given (empty) flow. In drawing a residual graph, to show a forward edge with capacity x and a backward edge with capacity y, annotate the original edge $\vec{x}; \overleftarrow{y}$.

[4 marks]

(b) What is the bottleneck edge of the path (s, v_1, v_3, v_5, t) in the residual graph you have given in answer to part (a) ?

[2 marks]

(c) Show the network with the flow (s, v_1, v_3, v_5, t) that results from augmenting the flow based on the path of the residual graph you have given in answer to part (a).

[3 marks]

(d) Show the residual graph for the network flow given in answer to part (c).

[4 marks]

(e) 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 (d) ?

[2 marks]

(f) Show the network with the flow that results from augmenting the flow based on the path (s, v_3, v_4, t) of the residual graph you have given in answer to part (d).

[3 marks]

(g) Show the residual graph for the network flow given in answer to part (f).

[4 marks]

(h) What is the bottleneck edge of the path $(s, v_2, v_3, v_1, v_4, t)$ in the residual graph you have given in answer to part (g) ?

[2 marks]

(i) Show the network with the flow that results from augmenting the flow based on the path $(s, v_2, v_3, v_1, v_4, t)$ of the residual graph you have given in answer to part (g).

[3 marks]

(j) Show the residual graph for the network flow given in answer to part (i).

[4 marks]

(k) Show the final flow that the Ford-Fulkerson Algorithm finds for this network, given that it proceeds to completion from the flow rates you have given in your answer to part (i), and augments flow along the edges (s, v_1, v_3, t) and (s, v_2, v_5, t).

[4 marks]

(l) Identify a cut of the network that has a cut capacity equal to the maximum flow of the network.

[5 marks]

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

END.