Candidate Number

G6017

THE UNIVERSITY OF SUSSEX

BSc and MComp SECOND YEAR EXAMINATION
January 2017 (A1)

PROGRAM ANALYSIS

Assignment Project Exam Help

Assessment Period: January 2017 (A1)

https://powcoder.com

DO NOT TURN OVER UNTIL INSTRUCTED TO BY THE CHIEF INVIGILATOR

Add WeChat powcoder

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 any answer books/answer sheets, used or unused, will be collected from you before you leave the examination room.

1. (a) Give the asymptotic running time of each of the following algorithms. Specify tight bounds when possible, and where appropriate, consider the algorithm's worst-case and best-case running time. You will not receive marks without a valid justification of your answer.

   All four algorithms take 3 integer sequences of length n as input, and note that $a_i \cdot b_j \cdot c_k$ denotes the product of the three integers a_i, b_j and c_k.

   i. AlgorithmA $(\langle a_1, \ldots, a_n \rangle, \langle b_1, \ldots, b_n \rangle, \langle c_1, \ldots, c_n \rangle)$:

```
total ← 0                                   (1)
for i ← 1 to n                              (2)
    total ← total + a_i · b_i               (3)
for i ← 1 to n                              (4)
    total ← total + a_i^2 · c_i             (5)
return total                                (6)
```

   [5 marks]

   ii. AlgorithmB $(\langle a_1, \ldots, a_n \rangle, \langle b_1, \ldots, b_n \rangle, \langle c_1, \ldots, c_n \rangle)$:

```
total ← 0                                   (1)
for i ← 1 to n                              (2)
    for j ← 1 to n                          (3)
        for k ← 1 to n                      (4)
            total ← total + a_i · b_j · c_k (5)
return total                                (6)
```

   [5 marks]

   iii. AlgorithmC $(\langle a_1, \ldots, a_n \rangle, \langle b_1, \ldots, b_n \rangle, \langle c_1, \ldots, c_n \rangle)$:

```
total ← 0                                   (1)
for i ← 1 to n                              (2)
    for j ← i to n                          (3)
        for k ← j to n                      (4)
            total ← total + a_i · b_j · c_k (5)
return total                                (6)
```

   [5 marks]

   iv. AlgorithmD $(\langle a_1, \ldots, a_n \rangle, \langle b_1, \ldots, b_n \rangle, \langle c_1, \ldots, c_n \rangle)$:

```
total ← 0                                   (1)
for i ← 1 to n                              (2)
    if a_i ≥ 0 then                         (3)
        for j ← i to n                      (4)
            if a_j ≥ 0 then                 (5)
                for k ← j to n              (6)
                    if a_k ≥ 0 then         (7)
                        total ← total + a_i · b_j · c_k   (8)
return total                                (9)
```

   [5 marks]

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

(b) Consider two algorithms: Algorithm 1 and Algorithm 2 that both solve the same problem P. The following facts about these algorithms have been established.

Fact 1: The running time of Algorithm 1 is $O(n^5)$.

Fact 2: The running time of Algorithm 2 is $O(n^3)$.

Fact 3: The running time of Algorithm 1 is $\Omega(n^3)$.

Fact 4: The running time of Algorithm 2 is $\Omega(n^3)$.

For each of the following statements, indicate whether the statement is definitely true, possibly true, or definitely false. You must fully explain your answer. You will not receive any marks for an answer that does not include a valid explanation.

i. Since Algorithm 1 and Algorithm 2 solve problem P, it must be the case that they produce precisely the same output for each instance of problem P.

[5 marks]

Assignment Project Exam Help

ii. The running time of Algorithm 1 is $\Theta(n^3)$.

[5 marks]

https://powcoder.com

iii. The running time of Algorithm 2 is $\Theta(n^3)$.

[5 marks]

Add WeChat powcoder

iv. When choosing an algorithm that solves problem P, Algorithm 2 is a better choice than Algorithm 1.

[5 marks]

v. The best-case running time of Algorithm 1 is $O(n^5)$.

[5 marks]

vi. $\Omega(n^3)$ is a lower bound for the running time for problem P.

[5 marks]

Turn over/

2. (a) Give a precise formulation of the single-source, shortest path problem for weighted graphs. In particular, you should specify what the inputs to the problem are, and describe what form and value the outputs take.

[5 marks]

(b) Weighted graphs are used to model a wide variety of problem scenarios. For example, a transportation network where weights of edges correspond to journey times.

In general, weighted graphs can include edges with edge weights that are negative. Describe a scenario that one might want to model with a graph where it would be plausible that negative weights would arise.

[5 marks]

(c) A graph is said to have a negatively weighted cycle when the sum of the weights of the edges involved in the cycle is negative. Why is it not possible to solve the single-source, shortest path problem for graphs that contain negatively weighted cycles?

[5 marks]

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

(d) Dijkstra's algorithm is not guaranteed to work correctly when given a graph containing negative weights. This problem arises even when there are no negatively weighted cycles in the graph.

Give an example of a graph with three vertices containing negative weights, but no negatively weighted cycles, for which Dijkstra's Algorithm will fail to produce a valid output. Clearly explain where Dijkstra's algorithm goes wrong when given your example as input.

For reference, Dijkstra's single-source, shortest path algorithm is shown on the next page.

[10 marks]

Algorithm Dijkstra(G, w, s) :

/* $G = (V, E)$ *is a graph with vertices* V *and edges* E*;*
w *is a function assigning the weight* $w(e)$ *to edges* $e \in E$*;*
and s *is the designated source vertex, an element of* V*.* */

/* *Initialise* δ */

for $v \in V$ let $\delta(v) = \begin{cases} 0 & \text{if } v = s \\ \infty & \text{otherwise} \end{cases}$ (1)

/* $\delta(v)$ *is the estimated (shortest) distance from* s *to* v*.*
This estimate will be the true value once algorithm is complete. */

let Q be priority queue of elements of V prioritised by least $\delta(v)$ (2)

let A be the empty set (3)

/* *The set* A *holds all the vertices that have been removed from* Q*.*
Once a vertex v *is in* A *we know that* $\delta(v)$ *is the true shortest*
distance from s *to* v*.* */

/* *All initialisation complete, now the main loop of the algorithm* */

while Q is not empty (4)

remove v from front of priority queue Q (5)

add v to A (6)

for each $\{ v, u \} \in E$ where $u \notin A$ (7)

/* *Check to see if there is a shorter route from* s *to* u *via* v */

if $\delta(u) > \delta(v) + w(v, u)$ then (8)

/* *update* $\delta(u)$ */

let $\delta(u) = \delta(v) + w(v, u)$ (9)

return δ (10)

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Question continued over

(e) The shortest path problem for *all* weighted graphs not containing negatively weighted cycles can be solved by the following dynamic programming algorithm, known as the BellmanFord Algorithm.

BellmanFord$(G = (V, E), w, s, t)$:

/* $G = (V, E)$ *is a graph with vertices* V *and edges* E*;*
w *is a function assigning edge weights to edges in* E*;*
and s *and* t *are vertices in the* V*.*

The algorithm will find the shortest path in G *from* s *to* t*.* */

/* $B(i, v)$ *will store the shortest distance from* v *to* t *over paths containing at most* i *edges* */

/* *Initialise* $B(0, x)$ *for all vertices* x*.* */

let $B(0, t) \leftarrow 0$ (1)
let $B(0, v) \leftarrow \infty$ for each $v \in V - \{t\}$ (2)

/* *Now complete the rest of the entries in table* B */

for $i \leftarrow 1$ to $|V| - 1$ (3)
for each $v \in V$ (4)

/* *Find the smallest value to put in* $B(i, v)$ */

/* *Start with* $B(i - 1, v)$ */

$B(i, v) \leftarrow B(i - 1, v)$ (5)
for each $\{ v, u \} \in E$ (6)

/* *Looking for a shorter path which starts with edge* $\{ v, u \}$ */

if $B(i, v) > w(v, u) + B(i - 1, u)$ then (7)

/* *Found a shorter route via* u */

$B(i, v) \leftarrow w(v, u) + B(i - 1, u)$ (8)

return $B(|V| - 1, s)$ (9)

i. Dynamic programming algorithms such as the BellmanFord Algorithm shown above maintain a table in which the solution to a space of subproblems can be solved.
In the case of the BellmanFord Algorithm, when given a graph with n vertices and m edges, how many shortest-path subproblems does the algorithm solve in total? Explain your answer.

[5 marks]

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Question continued over

ii. For a given i and v, where $1 \leq i \leq |V| - 1$ and $v \in V$, give a general characterisation of the number of steps that will be required to complete the entry $B(i, v)$. This question is asking for the running time of the *body* of the loop that starts on line 4, so you should refer to lines 5, 6, 7 and 8 of the above algorithm. Fully explain your answer.

[5 marks]

iii. Discuss the overall running time of the BellmanFord Algorithm. You should make reference to your answer to Question 2(e)ii. If appropriate you should consider both the best- and worst-case running times.

[5 marks]

iv. For graphs that do not contain any negative weights, is it more efficient to use Dijkstra's Algorithm or the BellmanFord Algorithm?

[5 marks]

v. Discuss what would happen if the BellmanFord Algorithm was given a weighted graph that contained a negatively weighted cycle.

[5 marks]

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

3. The question considers the Ford-Fulkerson Algorithm.

MaximumFlow$(G, s, t, c,)$:

/* $G = (V, E)$ *is a graph with vertices* V *and edges* E*;*
s *and* t *are vertices in the* V*;*
and $c(e)$ *is a function giving the capacity of edge* e *for all* $e \in E$*;*
The algorithm will find the maximum flow f *from* s *to* t*.* */

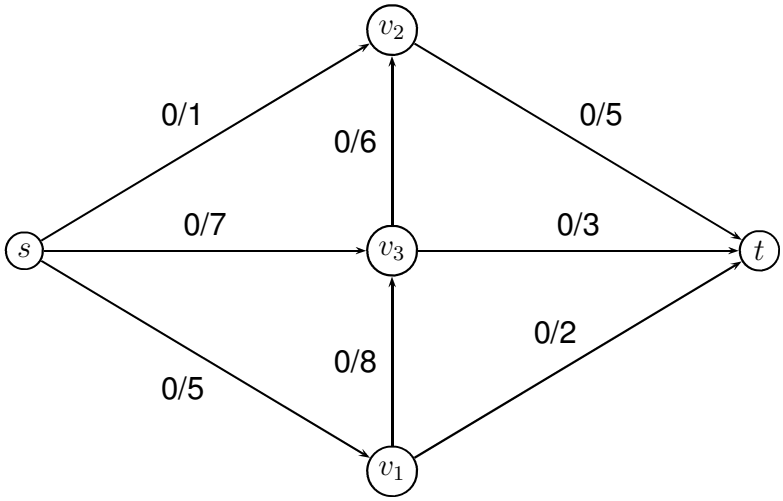
```
let f(e) = 0 for all e ∈ E                                   (1)
while there is a path from s to t in the residual graph G_f  (2)
    let p be a simple path from s to t                       (3)
    f ← augment(G, f, p)                                     (4)
    let G_f be the new residual graph associated with f      (5)
return f                                                     (6)
```

augment(G, f, p):

```
let the bottleneck edge in p have capacity x                 (1)
for each edge (u, v) in the path p                           (2)
    if (u, v) is a forward edge then                         (3)
        f(u, v) ← f(u, v) + x                                (4)
    if (u, v) is a backward edge then                        (5)
        f(v, u) ← f(v, u) − x                                (6)
return f                                                     (7)
```

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

We will look at how this algorithm would apply to the following network. 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.



Question continued over

(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 $\overrightarrow{x}\overleftarrow{y}$.

[5 marks]

(b) What is the bottleneck edge of the path (s, v_3, v_2, t) in the residual graph you have given in answer to Question 3a?

[5 marks]

(c) Show the network with the flow that results from augmenting the flow based on the path (s, v_3, v_2, t) of the residual graph you have given in answer to Question 3a.

[5 marks]

Assignment Project Exam Help

(d) Show the residual graph for the network flow given in answer to Question 3c.

[5 marks]

https://powcoder.com

(e) What is the bottleneck edge of the path (s, v_2, v_3, t) in the residual graph you have given in answer to Question 3d?

Add WeChat powcoder

[5 marks]

(f) Show the network with the flow that results from augmenting the flow based on the path (s, v_2, v_3, t) of the residual graph you have given in answer to Question 3d

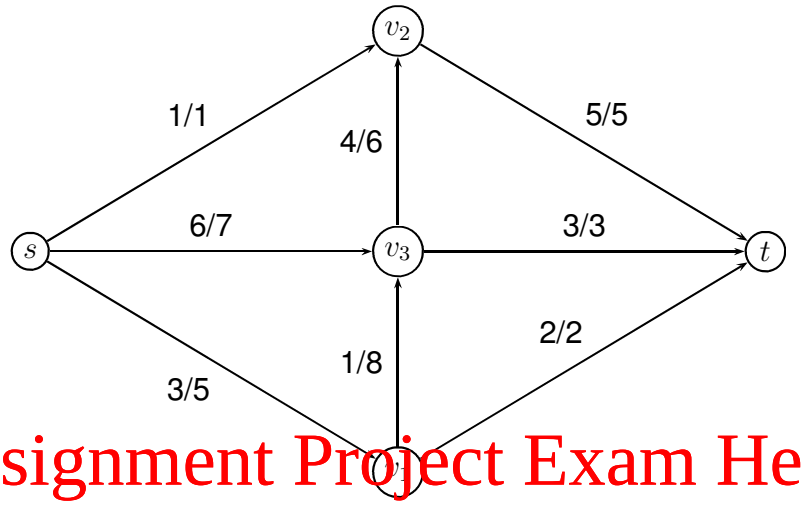
[5 marks]

(g) 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 answer to Question 3f.

[5 marks]

Question continued over

(h) Explain why the following flow would not be a valid answer to Question 3g.



Assignment Project Exam Help

https://powcoder.com

[5 marks]

(i) For each iteration of the while loop within the MaximumFlow algorithm shown above, a path p is selected from the residual graph G_f. There can be several such paths that the algorithm can choose. While the decision that the algorithm makes does not have an impact on whether a correct solution will be found, it can have a significant impact on the number of iterations of the while loop that are required to find a solution. For the network being considered in this question, what would be the best case in terms of the number of iterations required to find a solution. Fully justify your answer.

Add WeChat powcoder

[5 marks]

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

[5 marks]

End of paper