University of Glasgow

Friday 10 December 2021
09.00 am – 10.30 am
Duration: 1 hour 30 minutes
Additional time: 30 minutes
Timed exam – fixed start time

DEGREES OF MSc, MSci, MEng, BEng, BSc,MA and MA (Social Sciences)

ALGORITHMICS I (H)
COMPSCI4009

RUBRIC: Answer All Questions

This examination paper is worth a total of 60 marks.

1. Recall that the basic operations for transforming strings are:

- the insertion of a single character;
- the deletion of a single character;
- the substitution of one character for another.

Given these basic operations, the distance between two strings a and b of lengths m and n, is defined to be the smallest number of basic operations needed to transform a into b.

Letting $d_{i,j}$ denote the distance between the ith and jth prefix of the strings a and b respectively, then this distance is defined by the following recurrence relation:

$$d_{i,j} = \begin{cases} d_{i-1,j-1} & \text{if } a[i] = b[j] \\ 1 + \min\{d_{i,j-1}, d_{i-1,j}, d_{i-1,j-1}\} & \text{otherwise} \end{cases}$$

subject to $d_{i,0} = i$ and $d_{0,j} = j$ for all $i \leq m$ and $j \leq n$.

(a) Modify the above recurrence relation so that it instead computes the minimum cost of transforming the ith prefix of the string a into the jth prefix of b where:

- the insertion of a single character has cost 1;
- the deletion of a single character has cost 2;
- the substitution of one character for another has cost 3.

Hint: do not forget the base case.

[5]

(b) Using the recurrence relation from **(a)**, find the minimum cost of transforming the string a into string b where:

$$a = \texttt{saturday} \quad \text{and} \quad b = \texttt{sunday}.$$

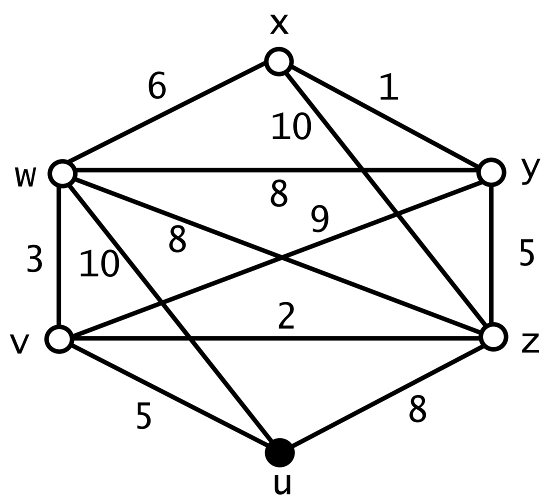
[6]

(c) Using the table from part **(b)** and the so-called traceback method, find the operations that yield the minimum cost of transforming a into b.

[4]

2. **(a)** Apply Dijkstra's shortest path algorithm to the graph below to find all shortest paths between the vertex *u* and all other vertices.

[8]




```
wt({u,v})=5
wt({u,w})=10
wt({u,z})=8
wt({v,w})=3
wt({v,y})=9
wt({v,z})=2
wt({w,x})=6
wt({w,y})=8
wt({w,z})=8
wt({x,y})=1
wt({x,z})=10
wt({y,z})=5
```

Include in your answer the steps performed by the algorithm through the changes to the distance and predecessor for each vertex of the graph.

(b) Assuming you have a graph with *n* vertices and *m* edges, discuss, in your own words, the advantages and disadvantages of using only a *basic heap* (priority queue) data structure *without pointers* over an array for storing the distances when implementing Dijkstra's shortest path algorithm in the *worst case* (i.e. using the big *O* notation in terms of *n* and *m*).

[7]

3. **(a)** Consider three decision problems P_1, P_2 and P_3. It is known that P_1 is in **P** and P_2 is NP-complete and P_3 is in **NP**. For each of the following statements, decide whether it is true or false. Include an explanation of your reasoning in each case.

1. P_3 is in **P** if there is a polynomial-time reduction from P_1 to P_3.
2. P_3 is in **P** if there is a polynomial-time reduction from P_3 to P_1.
3. P_3 is NP-complete if there is a polynomial-time reduction from P_2 to P_3.
4. P_3 is NP-complete if there is a polynomial-time reduction from P_3 to P_2.

[9]

(b) Consider three decision problems P_1, P_2 and P_3. It is known that P_1 is decidable and P_2 is undecidable. For each of the following statements, decide whether it is true or false. Include an explanation of your reasoning in each case.

1. P_3 is decidable if any instance of P_1 can be reduced to a instance of P_3.
2. P_3 is undecidable if any instance of P_3 can be reduced to a instance of P_2.
3. P_3 is undecidable if any instance of P_2 can be reduced to a instance of P_3.

[6]

4. **(a)** Describe, using a diagram or otherwise, a deterministic finite state automaton to recognise the language L_1 consisting of all strings over the alphabet $\Sigma = \{a, b\}$ that contains at least two b's and an a (in any order).

[4]

(b) Describe, using a diagram or otherwise, a deterministic finite state automaton to recognise the language L_2 consisting of all strings over the alphabet $\Sigma = \{a, b, c\}$ such that if there are precisely two c's in a row, i.e. two c's where any preceding or succeeding character is not a c, then the two c's are followed directly by either an a or or two b's.

This means that L_2 also includes any string that only contains a's and b's is in the language as well as any string where the number of c's in a row is always either one or three or more, as well as strings such as *abcca*, *ccbbab*, *ababccabb* and *bccbbcca*.

[5]

(c) Give a regular expression for the language L_2 described in part **(b)**.

[6]

 END OF QUESTION PAPER