UNIVERSITY OF WESTMINSTER

FACULTY OF SCIENCE & TECHNOLOGY

Department Title: Computer Science

Module: Algorithms: Theory, Design and Implementation

Module Code: 5SENG001W

Module Leader: Dr. E. Kapetanios
Date: 8th of May, 2017
Start: 10:00
Time allowed: 1 hour and 30 minutes

Instructions for Candidates:
You are advised (but not required) to spend the first ten minutes of the examination reading the questions and planning how you will answer those you have selected. You need to address ALL four sections.

DO NOT TURN OVER THIS PAGE
UNTIL THE INVIGILATOR INSTRUCTS YOU TO DO SO

SECTION A. [30 Marks, 2 x 15 correct {question, answer} pairs]

Please read carefully the following statements, which provide answers to the corresponding questions listed below as well. Answers and questions have been mixed up and the order of appearance does not imply the correct matching pair **{question, answer}**. Subsequently, you are asked to write down all correct matching pairs between questions and answers by using the indicating letters and numbers, respectively.

STATEMENTS/ANSWERS:

1. I can figure out whether I can go from a place A to a place B.

2. Implement the algorithms and measure the time the programs take to respond to the input by scaling the input data size, e.g., size of the network.

3. We want to compare the algorithms, in terms of their performance, no matter the amount of data received as input.

4. If, during the next execution of the algorithm, I am doubling the size of input data, the time spent for the algorithm to terminate is eight (8) times higher.

5. The total number N of elements to be multiplied within the input matrices.

6. *Mergesort* as this will never get worse than **n log n**.

7. The fact that it is in-place (uses only a small auxiliary stack), in contrast with the many copies of arrays requested by *Mergesort*.

8. In a worst case scenario, we are expecting **m(n-m+1)** comparisons to take place.

9. The key comparison A[j] < A[min].

10. (n (n-1)) / 2, where n is the number of elements to be sorted.

11. The Web, since we encounter pages that contain references (links) to other pages and we move from page to page by clicking on the links.

12. Calculating the allowed balance on each node of the tree.

13. Parent nodes with 3 pointers.

14. It brings the whole array into quasi already sorted elements such that the final round of *h=1* comparisons does not have to do too many insertions.

15. *Partitioning an* array into two subarrays by choosing a pivotal element, then keep on applying the same procedure recursively for sorting the resulted subarrays.

QUESTIONS:

A. What is the reason we analyse the performance of algorithms?

B. Which one of the algorithms, Quick-Union or Quick-Find, is more efficient as an algorithmic solution for the UNION-FIND problem?

C. What is the input data size we should be considering for an algorithm, which multiplies two input matrices?

D. What sort of answers you may be expecting from the railway network operators if this is run by the UNION-FIND algorithm?

E. What is the interpretation of algorithm performance being cubic (N^3)?

F. What is the basic operation of the *Selection Sort* algorithm?

G. What is the performance of *Selection Sort*?

H. What is the performance of the brute-force string (pattern) matching algorithm?

I. Which sorting algorithm shall we choose in order to optimise the performance of the *checking uniqueness* algorithm and making it better than the brute-force based approach?

J. Why does *Quicksort* appear to be more popular than *Mergesort* though they perform equally well?

K. Which is the key idea behind the *Quick S*ort algorithm?

L. Which is the key idea behind the *Shell S*ort algorithm?

M. What is the main feature of 2-3-4 trees defining their structure?

N. What is the main idea behind the balancing algorithm for AVL trees?

O. Which of our world technologies should be treated as a graph representation and algorithm?

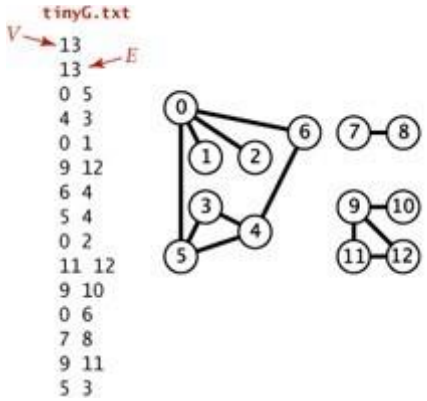
SOLUTION:

{A, 3}, {B, 2}, {C, 5}, {D, 1}, {E, 4}, **[10 marks]**

{F, 9}, {G, 10}, {H, 8}, {I, 6}, {J, 7}, **[10 marks]**

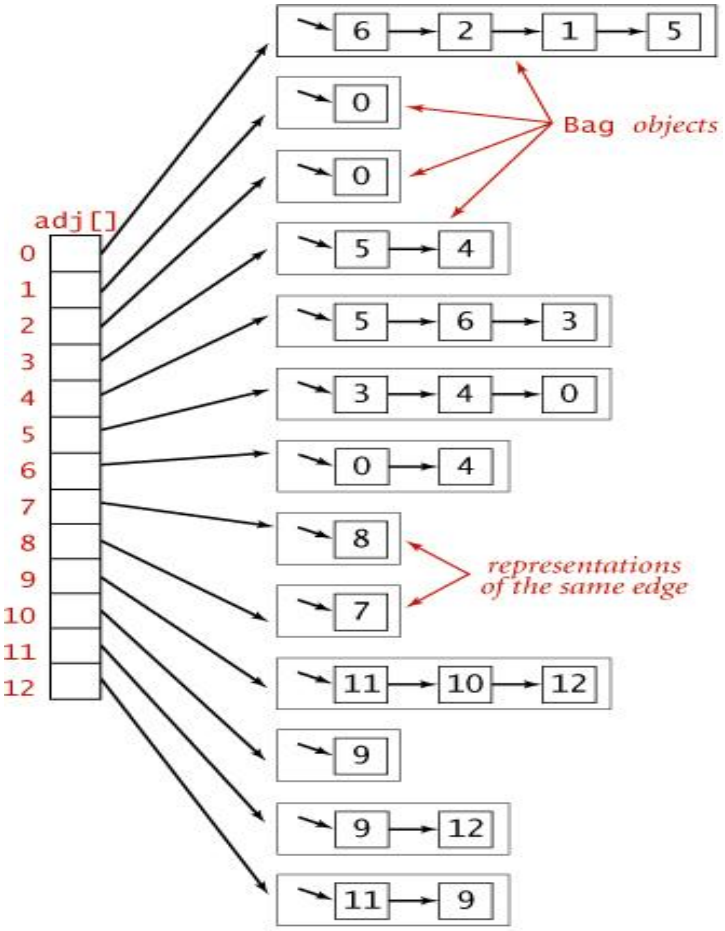
{K, 15}, {L, 14}, {M, 13}, {N, 12}, {O, 11} **[10 marks]**

SECTION B. [20 Marks]



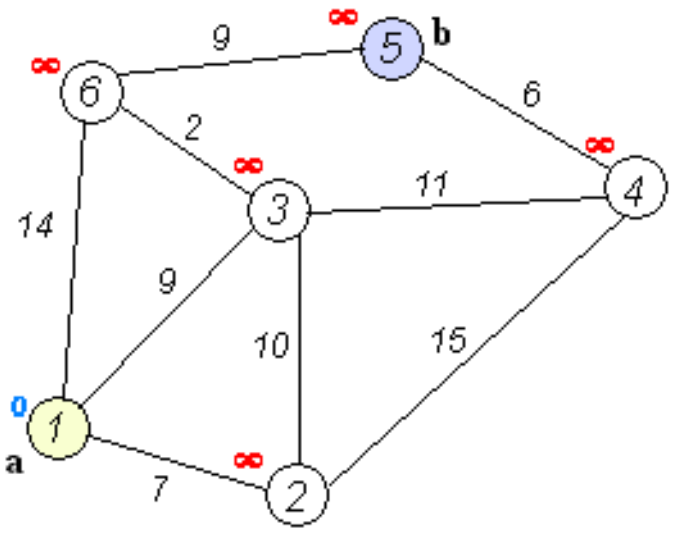
The figure above depicts an undirected graph, which has been generated with the tinyG.txt data on the left. Design and draw an adjacency list capable of representing the graph above.

SOLUTION: [20 Marks, 1.5 mark per correctly represented vertex as of the lists below, 0.5 mark for the array]



Adjacency-lists representation (undirected graph)

SECTION C. [25 Marks]



The above weighted graph has 6 interconnected vertices numbered from 1 to 6. The value between the two vertices is known as the edge cost between two vertices. For example, the edge cost between 1 and 6 is 14. Using the above graph and Dijkstra's algorithm,

1) determine the shortest path, in terms of lowest cost possible, from the source vertex 1 to destination vertex 5 **[5 Marks]**.
2) Justify your answer by writing down in the correct order of sequence all intermediary steps taken by Dijkstra's algorithm to calculate the shortest path from 1 to 5, including initial and end conditions of the graph. **[20 Marks]**

SOLUTION:

Shortest path is 1 – 3 – 6 – 5 (total cost: 20) **[5 Marks]**

Start condition:
Visited and current node: 1, Total Cost: 0 **[1 Mark]**
on all other nodes is INFINITY **[1 Mark]**

Visited node: 2, Total Cost: 7 **[2 Mark]**
Visited node: 6, Total Cost: 14 **[2 Mark]**
Visited node: 3, Total Cost: 9 **[2 Mark]**

Current node: 2
Visited node: 3, Total Cost: 17 (no update since 9 < 17) **[2 Marks]**
Visited node: 4, Total Cost: 22 (15+7) **[2 Marks]**

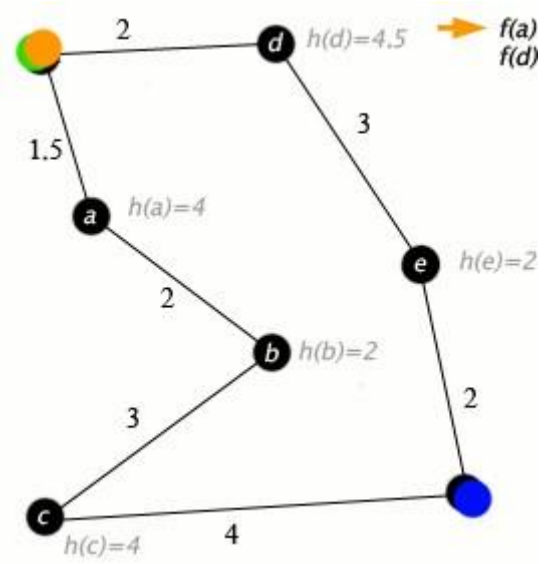
Current node: 3

Visited node: 4, Total Cost: 20 (update since 20 < 22) **[2 Marks]**
Visited node: 6, Total Cost: 11 (update since 11 (9+2) < 14) **[2 Marks]**

Current node: 6
Visited node: 5, Total Cost: 20 (11+9) **[2 Marks]**

Current node: 4
Visited node: 5, Total Cost: 26 (no update since 20 < 26) **[2 Marks]**

SECTION D. [25 Marks]



Given the road network depicted by the figure above, we apply an A* algorithm, where nodes are cities connected with roads at a cost measured in miles. The figure also includes the heuristic h(x) cost being used, which is the straight-line distance to the target point (dot on the right bottom corner). The starting point is the dot at the top-left corner. Based on this picture, answer the following questions:

a) What will be the elaborated shortest path from top-left to bottom-right corner cities?
[5 Marks]

b) Explain in terms of the f(x) = g(x) + h(x) function underpinning our A* algorithm what will be
   1) the nodes being visited;
   2) the calculated cost;
   3) the decisions made and at which point to take an alternative path.
[20 Marks]

SOLUTION:

a) Via nodes/cities (d) and (e) **[5 Marks]**

b)
f(a) = 1.5 + 4 = 5.5

f(d) = 2 + 4.5 = 6.5 **[3 Marks]**

f(b) = 3.5 + 2 = 5.5
f(d) = 2 + 4.5 = 6.5 **[3 Marks]**

f(c) = 6.5 + 4 = 10.5
f(d) = 2 + 4.5 = 6.5 **[4 Marks]**

This is the time to turn to the path via (d). **[6 Marks]**

f(c) = 6.5 + 4 = 10.5
f(e) = 5 + 2 = 7.5 **[4 Marks]**

END OF EXAM PAPER