Comp3010 Class Test 2

Ameer Karas

44948956

Question 1

a) This algorithm is a Las Vegas algorithm. This is because it will always find the correct result.

b) As the element used for comparison is selected randomly, the worst case scenario of computation would be n-1, or O(n). For example, say the number we are looking for is 1, our n value is 10 and the algorithm selects 9 as the element for the first comparison, 8 for the next, and so on. It would take n-1 comparisons to reach the correct answer, therefore O(n).

c) (Include working)

Q1

c) 1st comparison: $1/10$

2nd " : $1/9$

3rd " : $1/8$

$1/7$

$1/6$

$1/5$

$1/4$

$1/3$

$1/2$

so: $\frac{1}{10} \times \frac{1}{9} \times \frac{1}{8} \times \frac{1}{7} \times \frac{1}{6} \times \frac{1}{5} \times \frac{1}{4} \times \frac{1}{3} \times \frac{1}{2} \times 1 = 1/3628\,800$

so: $1/n!$

Probability of worst case is 1/n!.

d)

d) To get optimal result: if target x is 1, $n=10$, comparison element is 9 $\therefore$ 1 comparison

$1/10$ chance of this happening,

so $1/n$ is the probability

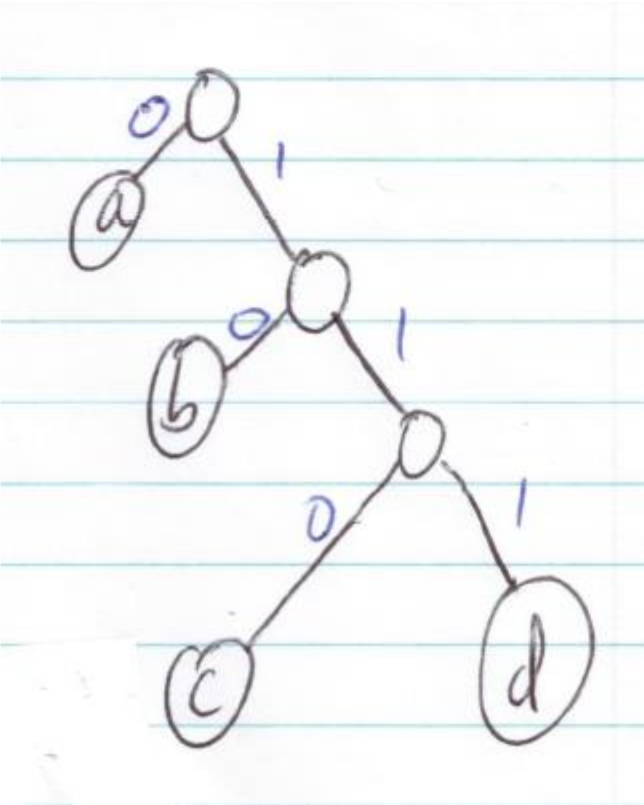
Question 2

Yes, this is possible if the 1-bit leads to a leaf of a tree that represents a character.

Q2

Character:	a	b	c	d
Frequency:	50	60	11	15

(handwritten example)



For the above example, the 1-bit string of 0 will lead to one of the characters, ‘a’.

Question 3

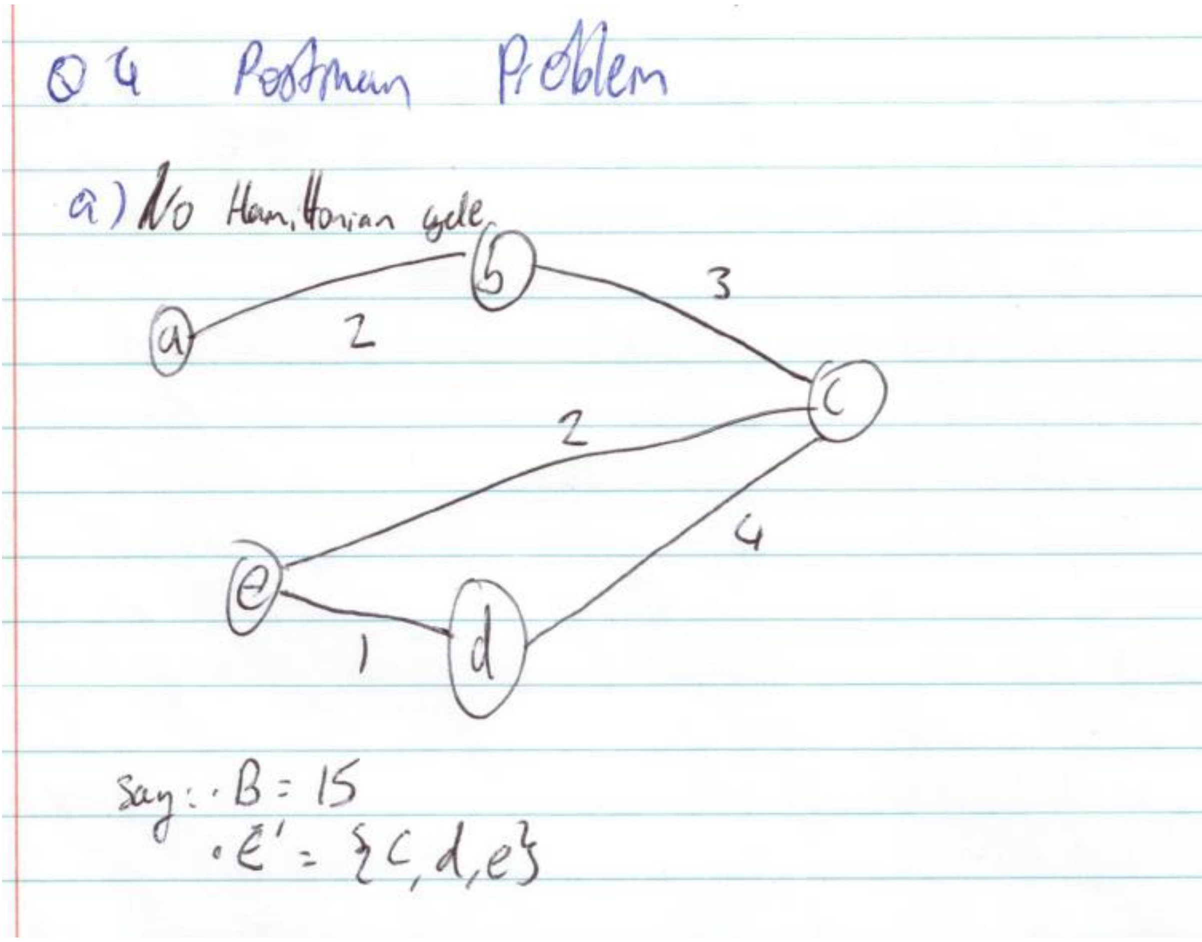
a) Every problem in NP can be reduced to problems in NP-complete. So, B is in NP, and B is also able to be reduced to A. This means that A is NP-complete.
b) A problem is in NP-complete if it is in NP and definitely hard, or at least as hard as any other problem in NP. A problem is NP-Hard if another problem in NP can be reduced in polynomial time to it. We know that B can be reduced to A, but we don’t know whether that reduction occurs in polynomial time. For this reason, A is not definitively in NP-complete.
c) P means that a problem can be solved in polynomial time. NP means that a problem is solvable in non-deterministic polynomial time. Both P and NP problems can be verified in polynomial time. If a problem B in NP can possibly be solved in P, then it is in P. There is no conclusive evidence as to whether NP problems are solvable in

polynomial time (i.e. NP = P), however this does not mean that it is impossible to solve B in polynomial time.

Question 4

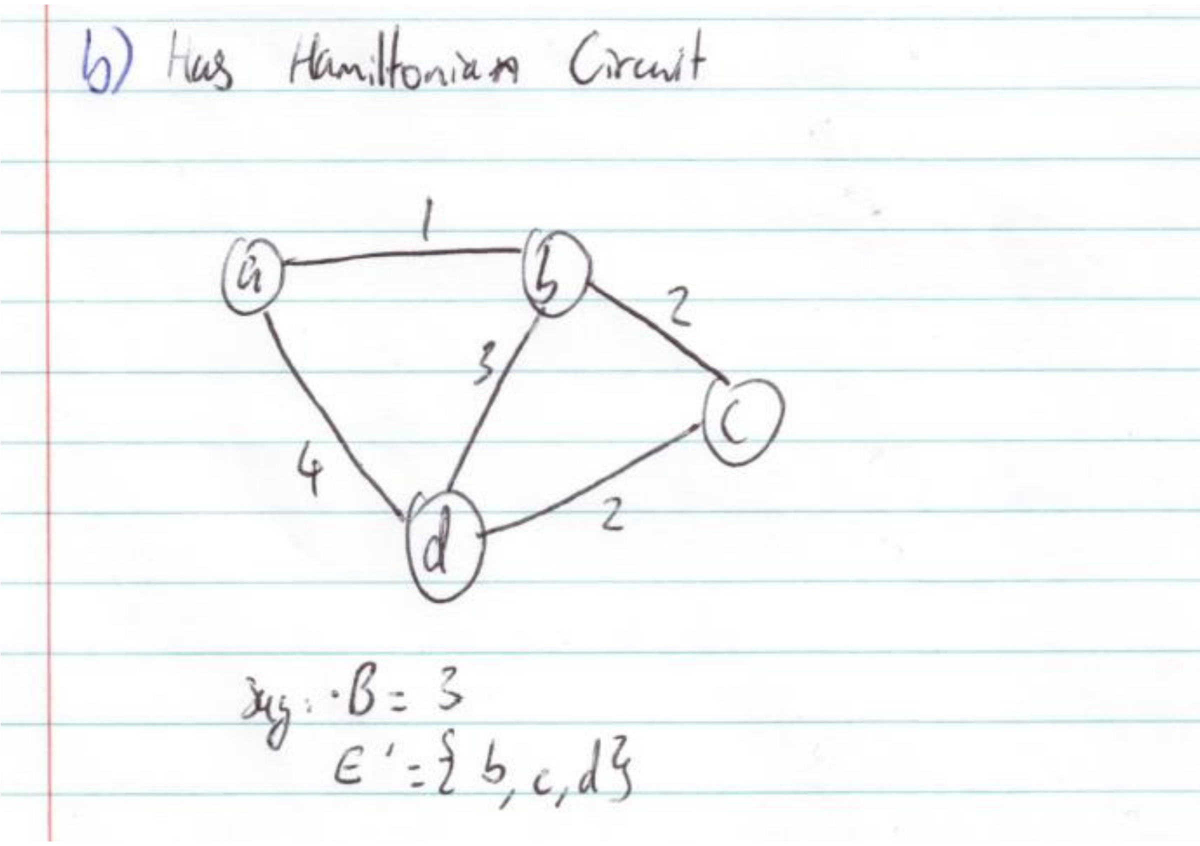
POSTMAN problem (decision problem): given an inclusive subset (E') of the edges (E) of a graph and a positive integer (B), can a circuit be found in a graph (G) that traverses E' at least once, and the sum of all the edges' weights is less than or equal to B.

a) (handwritten example).



Yes, if the graph has no Hamiltonian circuit but still has a cycle, as shown in the handwritten example. If we set B to 15 and E' = {c, d, e} then we have a circuit in G that traverses E' whose total weights are less than 15. Therefore, it will return a 'yes'.

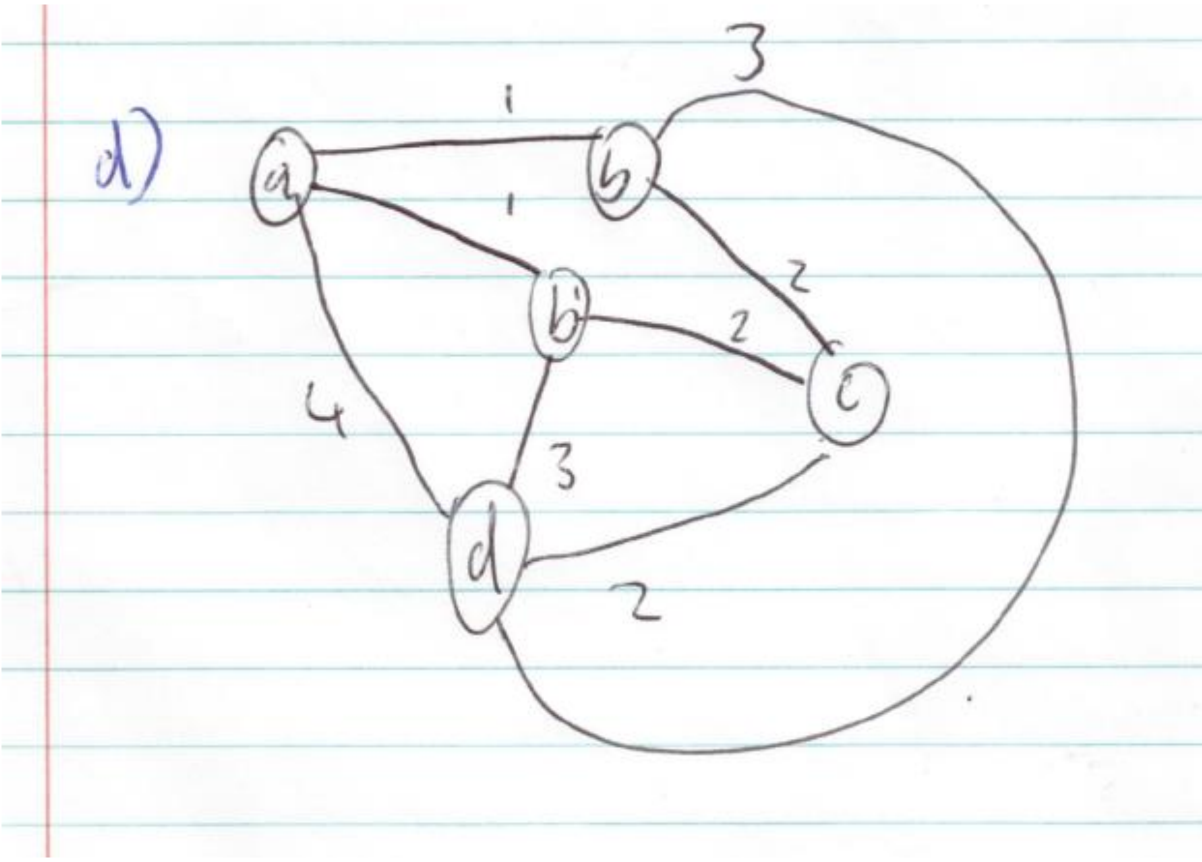
b) (handwritten example)



Whether or not POSTMAN returns a ‘no’ for this one will be determined by the value of B. If B is set to be too small then E’ will not be able to form a value from the given weights lower than B. Let’s do that and make B = 3 to demonstrate that it is possible for POSTMAN to return ‘no’ for this input, but only under specific circumstances. Here, the circuit defined by E’ would have a value of 7, resulting in ‘no’ being returned.

c) For this, our E’ value would be the path of the HAM-CIRCUIT. Our graph G would also need E’ to start and finish at the same vertex. So we now have the graph and E’ for HAM-CIRCUIT that we’ve had for POSTMAN.

d) This reduction is incorrect as it neglects the weights of the previous graph that would be used as part of the calculation to determine whether the output is a ‘yes’ or ‘no’; it ignores the B value and sum of the E’ edges. The standard POSTMAN graph would be represented as:



e) Given an algorithm for solving POSTMAN, we can solve HAM-CIRCUIT by:
    a. Constructing a complete graph G’ with an extra vertex
    b. Let B return 0 (as we are not looking for a set of edges whose weight are less than some value) for 2 adjacent vertices in the graph, and 1 otherwise

From here, we can solve the HAM-CIRCUIT problem exactly as we would the POSTMAN problem. The inputs would be G’, E’ and B= 0.

f) POSTMAN is in NP as it is easily verifiable, but cannot be solved in polynomial time, i.e. it is easy to see that the sum of E’ is less than B, but difficult to find E’ <= B.