|  |  |  |
| --- | --- | --- |
| BUE final logo  Informatics and Computer Science | **16CSCI01I**  **In - Class Test**  **2016 / 2017**  **Version 2** | |
| Module Title: **Analysis of Algorithms** | | |
| Module Leader **Dr. Mostafa Salama** | | Semester  **One** |
| Assessment Weight  **30% of the total course mark** | Due Date  **Announced on E-Learning** | |

|  |  |
| --- | --- |
| **Student ID:** |  |
| **Student Name:** |  |
| **Group No.** |  |

**Instructions to Students**

* The exam paper is **2** pages long, and is in **1** sections
* You should attempt **all** questions
* The allocation of marks is shown in brackets by the questions
* This examination is **1** hour long

---------------------------------------------------------

**Exam Instructions:**

* You should write down your answers within the test paper in the empty blanks below each question.

The course work is assessed as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Point number | Grade | Student grade |  |
| 1 | 8 marks |  |  |
| 2 | 6 marks |  |  |
| 3 | 6 marks |  |  |
| 4 | 10 marks |  | TA Signature |
| Total | 30 marks |  |  |

Feedback: The marks will be announced on the E-Learning with the model answer of the test.

1. Consider the following Brute-force algorithm that finds the distance between two closest points in a plane? [8 marks, 4 for each point]

**ALGORITHM** *BruteForceClosestPair(P)*

//Input: A list *P* of *n (n* ≥ 2*)* points *p*1*(x*1*, y*1*), . . . , pn(xn, yn)*

//Output: The distance between the closest pair of points

*d*←∞

**for** *i* ←1 **to** *n* − 1 **do**

**for** *j* ←*i* + 1 **to** *n* **do**

*d* ←min*(d, sqrt((xi*− *xj )*2 + *(yi*− *yj )*2*))*

**return** *d*

1. Show how to compute the number of instruction executed in this algorithm to find the results?
2. What is the complexity of this algorithm?
3. Set up and solve a recurrence relation for the number of times the following algorithm’s basic operation is executed. [6 marks]

Algorithm Q(n)

//Input: A positive integer n

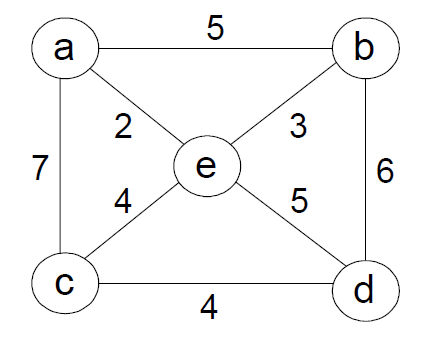
if n = 1 return 1

else return Q(n − 1) + 2 ∗ n − 1

1. State the Master Theorem then use it to find the O notation for the following recurrence relations. [6 marks]

T(n) = 4T(n/2) + 2n

1. Use Kruskal algorithm to find the minimum spanning tree of the following graph, clarify the steps of the algorithm in your solution [10 marks]



***Good Luck***

***Answer here:***

…………………………….…………………………………….…………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………………………….…………………………………….……………………………………………….…………………………………….………………………………………

|  |  |  |
| --- | --- | --- |
| BUE final logo  Informatics and Computer Science | **16CSCI01I**  **In - Class Test**  **2016 / 2017**  **Model Answer**  **Version 2** | |
| Module Title: **Analysis of Algorithms** | | |
| Module Leader **Dr. Mostafa Salama** | | Semester  **One** |
| Assessment Weight  **30% of the total course mark** | Due Date  **Announced on E-Learning** | |

1. Consider the following Brute-force algorithm that finds distance between two closest points in the plane? [8 marks, 4 marks for each]

**ALGORITHM** *BruteForceClosestPair(P)*

//Input: A list *P* of *n (n* ≥ 2*)* points *p*1*(x*1*, y*1*), . . . , pn(xn, yn)*

//Output: The distance between the closest pair of points

*d*←∞

**for** *i* ←1 **to** *n* − 1 **do**

**for** *j* ←*i* + 1 **to** *n* **do**

*d* ←min*(d, sqrt((xi*− *xj )*2 + *(yi*− *yj )*2*))*

**return** *d*

1. Show how to compute the number of instruction executed in this algorithm to find the results?
2. What is the complexity of this algorithm?

**Model Answer**:

1. The number of computations with respect to size of the problem which is n is as follow:

= 2 \* [(n-1)+ (n-2)+…+1]

= (n-1)\*n

1. The complexity of this algorithm is O(n2)
2. Set up and solve a recurrence relation for the number of times the following algorithm’s basic operation is executed. [6 marks]

Algorithm Q(n)

//Input: A positive integer n

if n = 1 return 1

else return Q(n − 1) + 2 ∗ n − 1

**Model Answer**:

The recurrence relation

M(n) = M(n − 1) + 2,

M(1) = 0.

The solution of the recurrence

M(n) = M(n − 1) + 2

= [M(n − 2) + 2] + 2 = M(n − 2) + 2 + 2

= [M(n − 3) + 2]+2+2 = M(n − 3)+2+2 + 2

= ...

= M(n − i) + 2i

= ...

= M(1) + 2(n − 1)

= 2(n − 1).

1. State the Master Theorem then use it to find the O notation for the following recurrence relations. [6 marks]

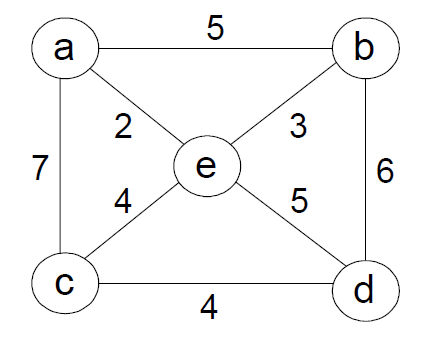
T(n) = 4T(n/2) + 2n

**Model Answer**:

f(n) = a f(n/b) + cnd

*a =* 4, *b* = 2 ⇒ Master theorem cannot be applied on this recurrence

1. Use Kruskal algorithm to find the minimum spanning tree of the following graph, clarify the steps of the algorithm in your solution [10 marks]



**Model Answer**

**Procedure** ***Kruskal (V***, ***E***)

**Begin**

***T***= {};

***n*** = |***V***|

**1. *Sort*** the edges of ***E*** in an ascending order according to weight ***w****;*

**2. While |T| < *n***

Remove (***u***,***v***)edge of lowest weight ***w*** from ***E***

**if**( ((***x***,***u***)edge ∈ ***T*** ) *&&* ((***y***,***v***)edge ∈ ***T***) )

disregard (***u***,***v***)edge

**else**

add (***u***,***v***)edge to ***T***

return ***T***;

**End *Kruskal*;**

