Name : Fadi Alahmad Alomar

ID : 1200180049

# Aasignment 1

## Question 1:

to know the asymptotic relation between and we need to find if it is equal to $\infin$ then if it is equal to then $f(n) \in \Omicron (g(n))$ if it is equal to a number then

* ,

$\lim\_{n \to \infin}{\frac{100n+\log{n}}{n + (\log{n})^2}} = \lim\_{n \to \infin}{\frac{100+\frac{1}{n}}{1 + \frac{2\log{n}}{n}}} = \frac{100}{1} = 100$

$ f(n) (g(n))$

* ,

$\lim\_{n \to \infin}{\frac{\log{n}}{\log{n^2}}} = \lim\_{n \to \infin}{\frac{\lim{n}}{2\lim{n}}} = \lim\_{n \to \infin}{\frac{1}{2}} = \frac{1}{2}$

$ f(n) (g(n))$

* ,

$*{n }{} =* {n }{( \* )} = *{n }{} =* {n }{} $

$\lim\_{n \to \infin}{\frac{n}{3(\log{n})^2}} = \lim\_{n \to \infin}{\frac{1}{\frac{6\log{n}}{n}}} = \lim\_{n \to \infin}{\frac{n}{6\log{n}}} = \lim\_{n \to \infin}{\frac{1}{\frac{6}{n}}} = \lim\_{n \to \infin}{\frac{n}{6}} = \infin$

$ f(n) (g(n))$

* ,

for this exercise we will try to show that after a specific . it is easy to show that if $ n $ then $ n$ as if we put in the base term of we get if we put we will find that the expression is evaluated to , were is some constant and . when .

$ f(n) (g(n))$

* ,

$ *{n }{} =* {n }{( \*)} = *{n }{} =* {n }{} = *{n }{} =* {n }{} = *{n }{(\* )} =* {n }{} = $

$ f(n) (g(n))$

* ,

prove by induction that $f(n) \in \Omicron(g(n))$ we need to prove that base step: induction hypothesis , R.T.P.

1. in order to reach we need to multiply by and when this term is smaller than 3
2. in order to get we need to multiply by 3 from 1 and 2 if we multiply the left side of by and the left side by the equality should not change which proves $f(n) \in \Omicron(g(n))$

* ,

we are doing asymptotic analysis is to the basee of $\lim\_{n \to \infin}{\frac{\sqrt{n}}{\sqrt{n}}} = \lim\_{n \to \infin}{1} = 1$

$ f(n) (g(n))$

## Question 2:

* in term of **unit cost**: the loop is done times and each loop we do $\Omicron(1)$ work thus this algorithm is $\Omicron(\log{m})$
* in term of **bit cost**: the loop is done times which is equal to and in each loop, we do a multiplication which takes $\Omicron(n)$ work the algorithm is $\Omicron(n\*n) = \Omicron(n^2)$

## Question 3:

we do BFS and put some indicators which alternates between two values that a node is visited and in each level search we enter we check if we have already visted the node that we are adding to the queue, we see if it is indicator is the opposite of the one we are currently in if it is the same then the graph is not a bipartite. if we visit all teh nodes and no two adjacent nodes have the same indicator then the graph is bipartite.

in the graph structure each node has an array that include it is neighbours

root <- Graph.root # the root node of the graph  
myHash <- Hash()  
myQueue <- Queue()  
FUNCTION Bipartite(root):  
 c <- -1  
 myHash[root] <-c  
 while myQueue.size != 0 do:  
 current <- myQueue.pop()  
 for i=0,i<current.neighbours.size do:  
 if myHash[current.neighbours[i]] do:  
 if myHash[current.neighbours[i]] != c\*-1 do:  
 return False  
 else do:  
 continue  
 else do:  
 myHash[current.neighbours[i]] = c\*-1  
 end  
 myQueue.push(current.neighbours[i])  
 end  
 c <- c\*-1  
 end  
 return True