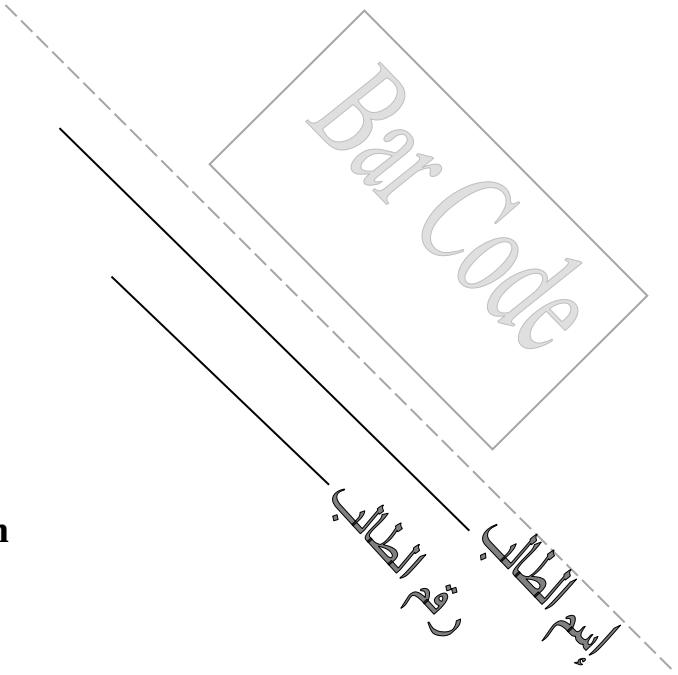


Cairo University
Faculty of Computers and Information

Final Exam

Department: SWEP/NTP /BIOP

Course Title: Algorithms Analysis & Design
Course Code: NCS314 / CS316/ BCS316
Semester: 1
Instructor: Dr- Basheer Abdel Fattah Youssef
Date: 09/01/2019
Exam Duration: 2 Hours

Question	Mark	Signature
One 10		
Two 10		
Three 7.5		
Four 7.5		
Five 7.5		
Six 10		
Seven 10		
Total Marks		

60

Total Marks in Writing: _____

Question (1) k-way-Merge Sort. Suppose you are given k sorted arrays, each with n elements, and you want to combine them into a single array of kn elements. Consider the following approach. Using the merge subroutine taught in lecture, you merge the first 2 arrays, then merge the 3rdgiven array with this merged version of the first two arrays, then merge the 4th given array with the merged version of the first three arrays, and so on until you merge in the final (kth) input array. What is the running time taken by this successive merging algorithm, as a function of k and n, show details of how to compute the running time**. (10 points)**

$2n + 3n + \dots kn = n\,(2 + 3 + \dots + k) = n\,((1 + 2 + 3 + \dots + k) - 1) = n\,(k(k+1)/2 - 1) = \theta(nk^2).$

Question (2) show amortized analysis for dynamic hash table which start by one size and extend 2 power i I in 0 1 ,2 4 8 ……. By aggregation methods **(10 points)**

Let $c_i =$ the cost of the i th insertion

$$= \begin{cases} i & \text{if } i-1 \text{ is an exact power of } 2, \\ 1 & \text{otherwise.} \end{cases}$$

i	1	2	3	4	5	6	7	8	9	10
$size_i$	1	2	4	4	8	8	8	8	16	16
c_i	1	1	1	1	1	1	1	1	1	1
		1	2		4				8	

$$\text{Cost of } n \text{ insertions} = \sum_{i=1}^{n} c_i$$

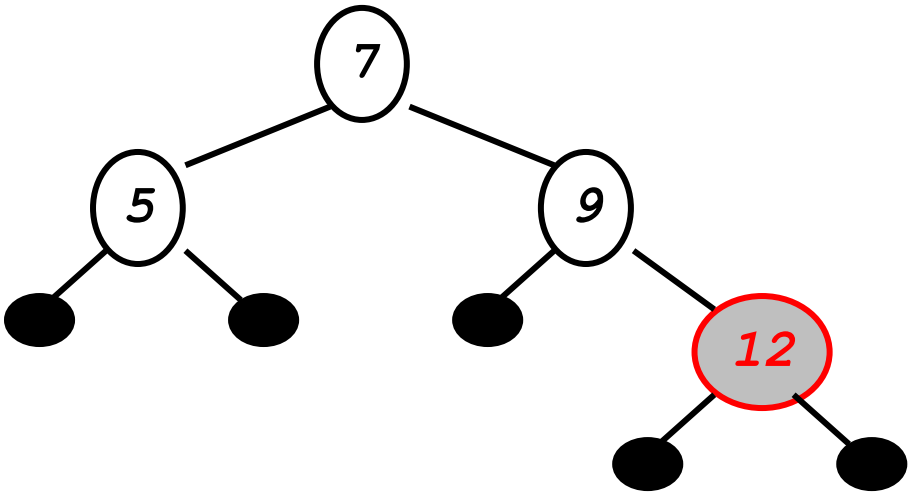
$$\leq n + \sum_{j=0}^{\lfloor \lg(n-1) \rfloor} 2^j$$

$$\leq 3n$$

$$= \Theta(n).$$

Thus, the average cost of each dynamic-table operation is $\Theta(n)/n = \Theta(1)$.

Question (3) color the following red-blact tree



Question (4) write a recursion function that compute a^b where a and b are positive integers. and compute its time complexity using master method. **(7.5 points)**

```
FastPower(a,b) :
            if b = 1
                 return a
            otherwise
                 c := a*a
                 ans := FastPower(c,[b/2])
             if b is odd
                 return a*ans
             otherwise return ans
end
```

T(n) =T(n/2) +1 = a=1 b = 2 c = 0 $n^0 \log n = \log n$

Question (5) Write an algorithm that finds both the smallest and largest numbers in a list of n numbers. Try to find a method that does at most 1.5 n comparisons of array items. Show the full analysis. **(7.5 points)**

First divide the n elements to two sub arrays n/2 according to compare the two adjacent values big go to first one and the small go to the second
This take n/2 comparisons
After that you can search for min in the second one take n/2
And search for max in the first one take n/2
Total =3n/2 = 1.5n

Question (6) use the greedy Algorithm and dynamic programming to solve the following Knapsack problem **(10 points) and determine which one is the best solution. W=5**

id	weight	value
1	1	10
2	2	15
3	3	14
4	1	12

Item/W	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	10	10	10	10	10
2	0	10	15	25	25	25
3	0	10	15	25	25	29
4	0	12	22	27	37	37

Items 1 ,2,4

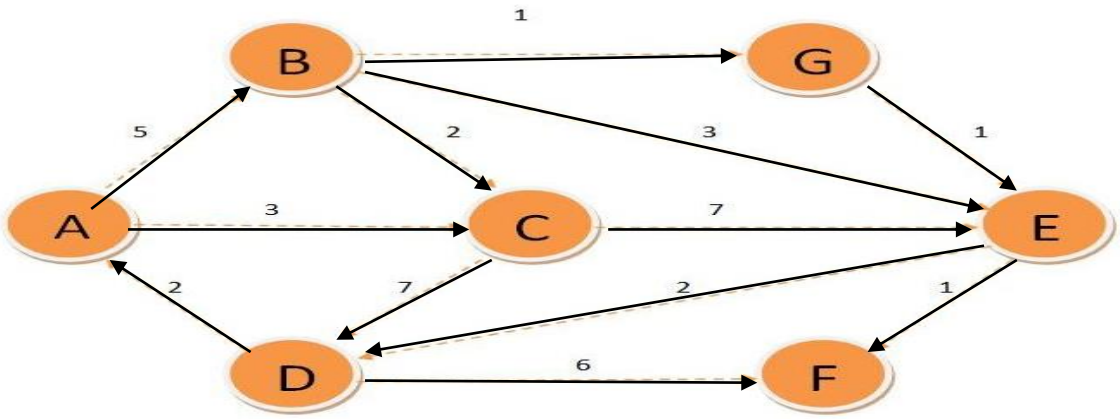
TIME DP = n*n optimum than greedy

Greedy if we chose max value first

15 + 14 = 29

DP best than Greedy

Question(7) Find the shotrest path for the following graph from node A to each nodes using Dijkstra's algorithm **(10 points)**



PARENT		A	A	E	G	E	B
#	A	B	C	D	E	F	G
	0	-	-	-	-	-	-
		5	3	-	-	-	-
		5		10	10	-	-
				10	8	-	6
				10	7	-	
				9		8	
				9			

ABG = 6

ABGEF = 8

ABGE =7

ABGED=9

AC = 3

AB = 5

Good Luck

Dr Basheer Youssef.