CS & IT ENGINEERING

Algorithms

Lecture No.- 03

1500 Series

By- Dr. Khaleel Khan sir

Recap of Previous Lecture

Topic ALGORITHMS 01

Topics to be Covered

Topic D and C

Topic Greedy Method

#Q. The time complexity to multiply two long integers of size n- digit represented in an array of size n, using divide & conquer strategy with 3-way split is-

A. $O(n)$

B. $O(n\log n)$

C. $O(n^{1.32})$

D. $O(n^2)$

LIM <Long Integer Multiplication>

1) Non-DC (School Method) : $O(n^2)$

2) (D and C)

2-way Split: $T(n) = 4T(n/2) + n$

3-way Split: $T(n) = 9 \cdot T(n/3) + n$

K-way Split: $T(n) = K^2 \cdot T(n/K) + n$

$O(n^2)$

$O(n^2)$

LIM

3) A.K optimiz. (DC)

2-way

$T(n) = 3 \cdot T(n/2) + n$

3-way

$T(n) = 8T(n/3) + n.$

K-way Split

$T(n) = (K^2 - 1)\, T(n/k) + n$

4) Toom & Cook :

2-way Split

$T(n) = 3T(n/2) + n$

3-way-Split

$T(n) = 5T(n/3) + n.$

K-way Split

$T(n) = (2K-1) \cdot T(n/3) + n$

#Q. The recurrence equation arises in solving which of the following problems:

MSQ

$T(n) = T(n/2) + 1,$

$T(1) = 1$

A Towers of Hanoi : $T(n) = 2T(n-1) + c$

B Merge sort : $T(n) = 2T(n/2) + n$

C Binary search : $T(n) = T(n/2) + c$

D Power of n (x^n) : $T(n) = T(n/2) + c$

D and C

#Q. Consider the following pseudocode of the recursive function goAgain.

```
//assume N is a positive integer
integer goAgain (integer a, integer N)
begin
  if(N ==1) then
        return a;
  else
        return (a + goAgain (a, N/2));
end
```

$T(n) = c \quad , \quad N = 1$

$= a + T(n/2) + b, \quad N > 1$

$T(n) = T(N/2) + d$

$O(\log N)$

What is the order of growth, as a function of N, of the number of times the function goAgain is recursively executed?

A $\Theta(\log N)$

B $\Theta(N)$

C $\Theta(N \log N)$

D $\Theta(N^2)$

E $\Theta(N^3)$

#Q. Which of the following algorithms has running time $\Theta(n^2)$ in the worst case but $\Theta(n \log n)$ on average?

A. Bubblesort : $n^2 + n^2$

B. Mergesort : $n \log n + n \log n$

C. Heapsort : $n \log n + n \log n$

D. Quicksort : $n^2 + n \log n$

#Q. Suppose that the splits at every level of quick sort are in the proportion $1-\alpha$ to α, where $0 < \alpha \le \frac{1}{2}$ is a constant. Minimum depth of a leaf in the recursion tree is approximately:

A. $-\frac{\log(n)}{\log(\alpha)}$

B. $\log(n\alpha)$

C. $n \log n$

D. $-\frac{n\log(n)}{\log(\alpha)}$

Let $\alpha = 1/3$

$n/3 \qquad 2n/3$

$T(n) = T(n/3) + T(2n/3) + n$

$T(n) \longrightarrow n$

$T(n/3) \quad T(2n/3) \longrightarrow n$

LHS $\qquad$ RHS

$\boxed{T(n/9)} \quad T(2n/9) \quad T(2n/9) \quad T(4n/9)$

$T(n/3^k)$

$\frac{n}{3^k} = 1 \Rightarrow k = \log_3 n$

For general value of "α"

$T((\alpha)^k \cdot n) = 1$

$\Rightarrow (\alpha)^k \cdot n = 1$

$(\alpha)^k = 1/n$

$= n^{-1}$

$k \cdot \log \alpha = -\log n$

$\therefore k = -\frac{\log n}{\log \alpha}$

#Q. Suppose that the splits at every level of quick sort are in the proportion $1-\alpha$ to α, where $0 < \alpha \le \frac{1}{2}$ is a constant. Maximum depth of a leaf in the recursion tree is approximately:

$$T\left((1-\alpha)^k \cdot n\right) =$$

$$(1-\alpha)^k \cdot n = 1$$

$$k = -\frac{\log n}{\log(1-\alpha)}$$

A. $\log n$

B. $-\log(n\alpha)$

C. $-\dfrac{\log(n)}{\log(1-\alpha)}$

D. $n \log n$

#Q. An algorithm's real-time readiness (RTR) ratio is defined as the ratio of its average-case running time to its worst-case running time.
Which of the following algorithm has an RTR ratio closest to 0?

$$\boxed{RTR = \frac{A \cdot C}{W \cdot C}}$$

A. Bubble sort $\frac{n^2}{n^2} = 1$

B. Heap sort $\frac{n \cdot \log n}{n \cdot \log n} = 1$

C. Insertion sort : $\frac{n^2}{n^2} = 1$

D. Merge sort

$\rightarrow \frac{n \cdot \log n}{n \cdot \log n} = 1$

E. Quick sort : $\frac{n \log n}{n^2} \sim 0$

#Q. Consider the following statements:

S_1: Quick sort has divide procedure but not combine. : ✓

S_2 : Number of comparisons in partition algorithm are same for best case and worst case : ✓

S_3 : Merge sort always has more comparisons than swaps. : ✗

Which of the following is/are correct?

A) S_1 and S_2 only

B) S_2 and S_3 only

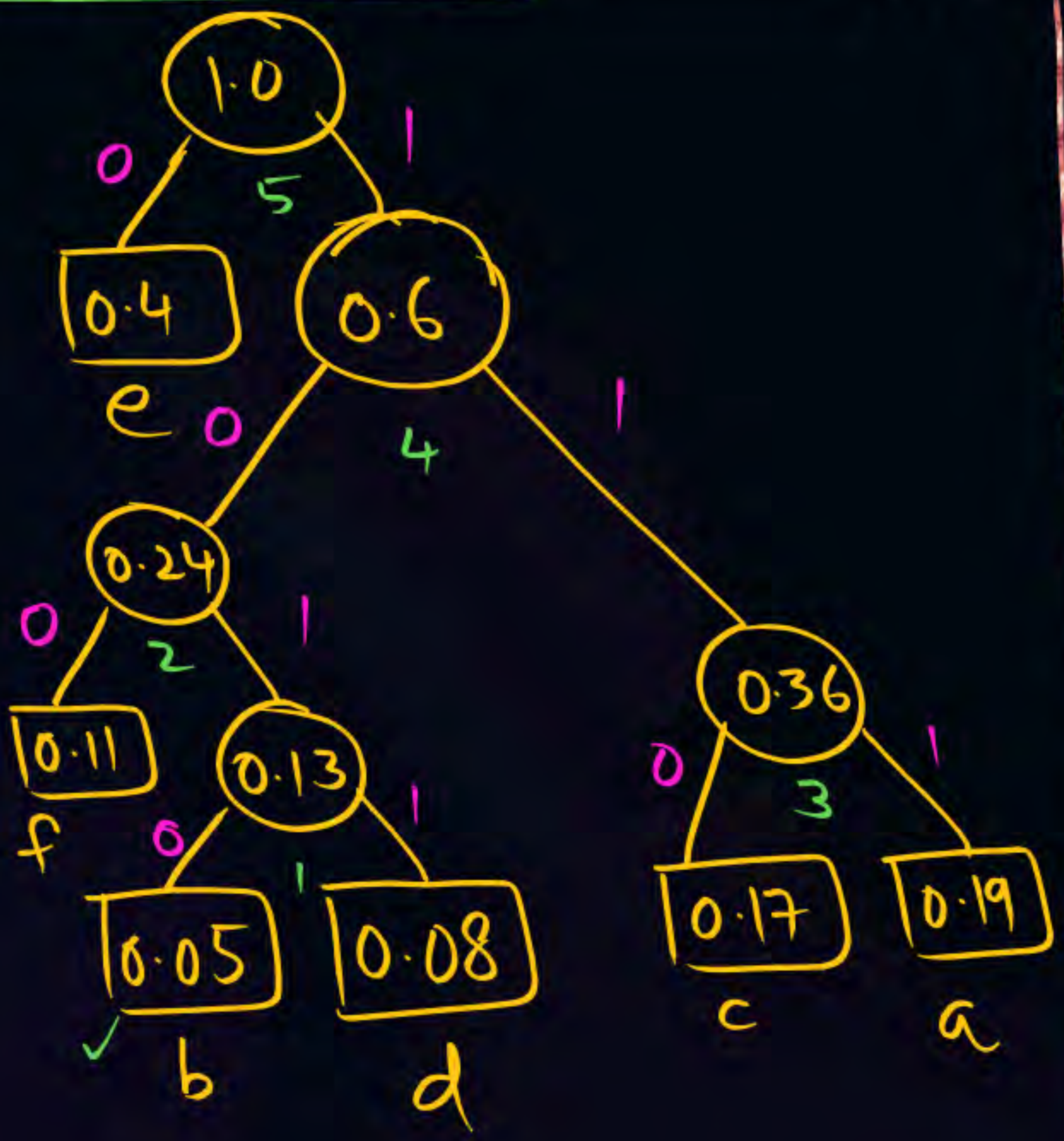
C) All are correct

D) None of the above

#Q. Consider a language L, defined as L=∑(a – f) with the relative frequencies respectively given as (0.19, 0.05, 0.17, 0.08, 0.4, 0.11), then average length of prefix binary code has 2.33 bits.

1

Huffman Coding:



a → 111 (3)
b → 1010 (4)
c → 110 (3)
d → 1011 (4)
e → 0 (1)
f → 100 (3)

$$\text{Avg bits} = \sum_{i=1}^{n} d_i \cdot q_i$$

$$= (3*0.19 + 4*0.05 + 3*0.17 + 4*0.08 + 1*0.4 + 3*0.11)$$

$=$

#Q. Two classical algorithms for finding a minimum spanning tree in a graph are Kruskal's algorithm and Prim's algorithm. Which of the following are the design paradigms used by these algorithms?

	Kruskal's algorithm	Prim's Algorithm
A	The greedy method	The greedy method
B	The greedy method	Dynamic programming
C	Dynamic programming	The greedy method
D	Dynamic programming	Divide and conquer
E	Divide and conquer	Dynamic programming

#Q. Consider the following possible data structure for a set of n distinct integers.

H/W

I. A min-heap

II. An array of length n sorted in increasing order

III. A balanced binary search tree

For which of these data structure is the number of steps needed to find and remove the 7^{th} largest element in $O(\log n)$ in the worst case?

A. I only

B. II only

C. I and II

D. I and III

E. II and III

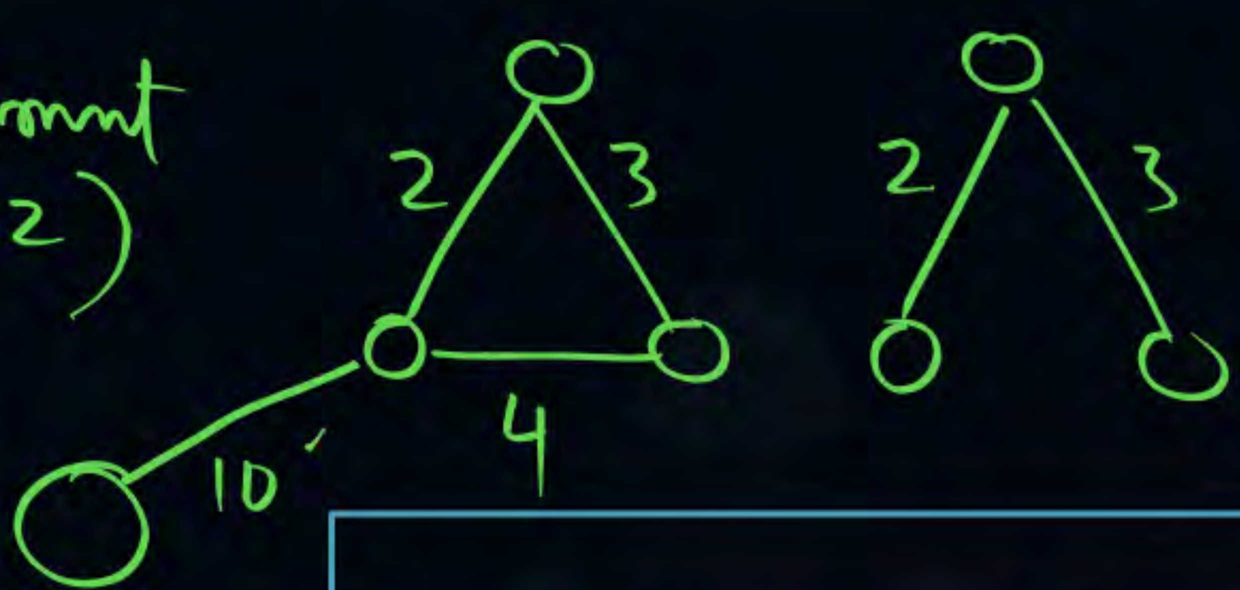
[NAT]

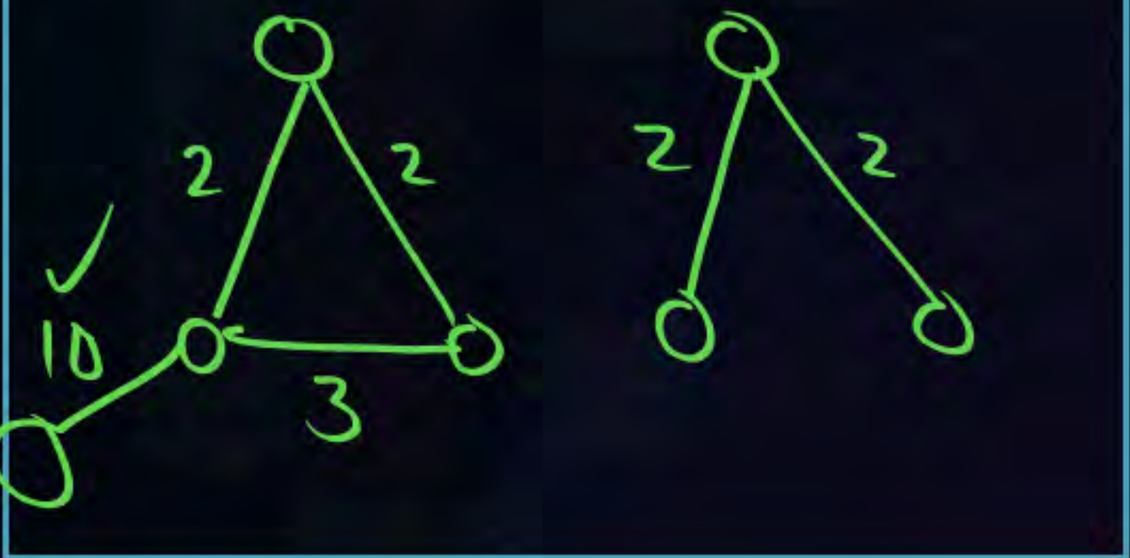
#Q. Consider the following statement with respect to a connected undirected graph G:

I : If G has a unique minimal spanning tree then all its edges are unique.

II : If G has all unique edges, then it always has a unique minimum spanning tree. T

The number of correct statement is/are 1 . (Stmnt 2)





#Q. Which of the following properties must be true of a Minimum Spanning Tree (MST) of a connected graph G with at least 3 edges?

I. The MST must contain the shortest edge of G.
II. The MST must contain the second-shortest edge of G.
III. The MST can never contain the longest edge of G.

A. None

B. I only

C. I and II only

D. I and III only

E. I, II and III

#Q. A hiker faces the 0/1 knapsack problem. There are 7 items to be packed into the knapsack, each with value v_i and weight w_i as shown in the following table.

2 3 5 6 1 4

0.42 2 1.6 1 0.5 2.5 1.16

i	1	2	3	4	5	6	7
Profit V_i	3	6	8	1	2	5	7
weight W_i	7	3	5	1	4	2	6

$\Rightarrow 29$

0/1 KNAP GM

H.A

H/W optimal Algo.

{0/1 KNAP D.P}

M

The knapsack, which is initially empty, can hold a maximum weight of 24, so some item(s) must be left behind, and fractions of items cannot be packed. The optimality criterion is to maximize the total value of the items that are placed in the knapsack. The hiker fills the knapsack one item at a time, using a heuristic algorithm that is greedy on value density, where the value density of an item is its value/weight ratio. When this heuristic algorithm is used, what is the total value of the items that are packed, and is this total optimal?

P_i/W_i

	Total Value	Optimal / Not Optimal
A	29	Optimal
B	29	Not optimal
C	30	Optimal
D	30	Not optimal
E	32	Optimal

[NAT]

<J.S.D>

#Q. Consider the following tasks with their profits and deadlines. What is the number of task than cannot be scheduled? (If the answer is 2 that means T_2 cannot be scheduled)

T_5

Tasks	Profit	Deadline
T_1	20	4
T_2	12	4
T_3	17	3
T_4	15	3
T_5	4	4
T_6	3	5

T_2	T_4	T_3	T_1	T_6	

0 1 2 3 4 5

#Q. The Character ‘p’ to ‘u’ have the set of frequencies based on the given table:

H/W

Character	Frequency
p	10
q	20
r	24
s	11
t	4
u	18

87

Frequencies :

Probab : < root : 1.0 >

$\frac{Freq_i}{\sum_{i=1}^{n} Freq_i}$ $p = 10/87$ $q = 20/87$

Which of the following is/are correct based on Huffman code?

A. For character sequence “t p t u q” corresponding hexadecimal code is “E F E 1”.

B. For character sequence “t p p q u” corresponding hexadecimal code is “E F F 1”.

C. For bit stream “10101100100” sequence of the character is “r r s q u”.

D. For bit stream “10101100001” sequence of character is “r r s u q”.

2 mins Summary

Topic	One	D and C
Topic	Two	Greedy Method
Topic	Three	
Topic	Four	
Topic	Five	

THANK - YOU