Introduction to Greedy Algorithms :→

Important Questions to be covered for Coding Interviews :→

1. (i) Minimum Number of Coins
2. (ii) Activity Selection Problem
3. (iii) Job Scheduling Problem
4. (iv) Chocolate Distribution Problem
   or
   Minimum Absolute Difference
5. (v) Minimum Cost of Ropes
6. (vi) Fractional Knapsack
7. (vii) 0/1 Knapsack
8. (viii) Huffman Encoding
9. (ix) Minimum number of jumps.
10. (x) Nikunj & Donuts.

* Minimum number of coins :

{1, 2, 5, 10, (20), (50), 100, 200, 500, 1000}

coins of different denominations

Target Value, V = 91

While { V >= coins(i) }

v >= 1000, 500, 200, 100 ×

>= 50

[50, 20, 20, 1] size

V -= coins(i)

91 − 50 = 41

41 − 20 = 21

21 − 20 = 1

1 − 1 = 0

*** Imp: Activity Selection Problem

* Given a set of activities with their start and finish times, select the max number of activities that can be completed by a single individual, with the condition that he/she can only perform one activity at a given time without overlapping.

Activity	Start	Finish	Sort (finish)	output
A1	5	7	→ A3 (1, 4)	A3 (1, 4)
A2	8	9	~~A6 (3, 5)~~	A1 (5, 7)
A3	1	4	~~A5 (0, 6)~~	A2 (8, 9)
A4	5	9	i A1 (5, 7)	
A5	0	6	A2 (8, 9)	
A6	3	5	~~A4 (5, 9)~~	

Minimum Cost of Ropes :→

arr = [4, 3, 2, 6]   cost = 0 + 5 + 9 + 15 = (29)

Sort = 2, 3, 4, 6   n log n

2, 3 = 2 + 3 = 5   ×

Sort [5, 4, 6]   n log n

= 4, 5, 6

4, 5   4 + 5 = 9   ×   Too Big

9, 6

Sort 6, 9   n log n   3.5 LPA

add 6 + 9 = 15

[4, 3, 2, 6] ⟶ O(1)

Min Heap

n log n < O(1) poll

"Dynamic Programming"

"Those who forget the past, are forced to repeat it."

* DP is the process of finding the solution of a bigger problem by solving smaller overlapping subproblems.

[Recursion Tree]

0, 1, 1, 2, 3, 5, (8), 13, 21, ...

0 1 2 3 4 5 6 7 8

$f(n) = f(n-1) + f(n-2)$

$f(0) = 0$

$f(1) = 1$

(i) Recursion

(ii) Memoization

(iii) Tabulation

(iv) Space optimⁿ