Greedy Algorithms continued !!!

*** Important :→ Q.555 → 21, 23, 24

1. Activity Selection Problem :→

Given a set of activities with their start & finish times, select the maximum number of activities that can be performed by a single person, assuming that a person can only work on a single activity at a given time.

→ I/P → {5,7}, {8,9}, {1,4}, {5,9}, {0,6}, {3,5}

O/P	Activity	Start Time	Finish Time
A1	✓ A1	1	4
	A2 ✗	3	5
	A3 ✗	0	6
A4	✓ A4	5	7
A5	✓ A5	8	9
(3)	A6	5	9

Steps : (i) Sort the activities according to their finish times in ascending order.

(ii) We select the activity which finishes earliest. For each activity if the start time is after or equal to the finish time of the selected activity, we select it.

(iii) The final selected activities are those which are selected without overlapping.

Dynamic Programming :→ (Recursion)

* Those who forget the past are forced to repeat it.

* Overlapping subproblems.

1. Recursion
2. dp[ ] → Memoisation
3. Tabulation

* 4. Space – Optimization

1D, 2D → LCS

1D DP, 2D DP "Abdul Bari" Pseudo code

Fibonacci Series :→ (Recursion Tree)

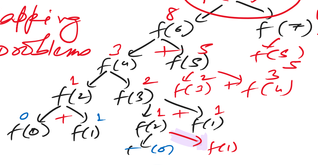
0, 1, 1, 2, 3, 5, 8, 13, 21 . . . . .

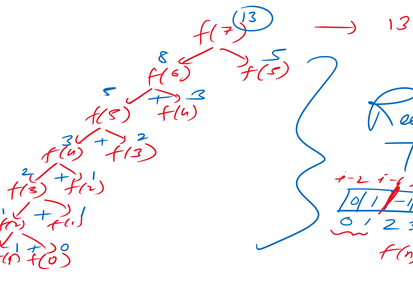
0 1 2 3 4 5

f(0) f(1)

f(n-2) f(n-1) f(n) (21)

Overlapping Subproblems





P1

0, 1, 1, 2, 3, 5, 8, 13, ...

P2 curr

$P_2 = 0$

$P_1 = 1$

$curr = p^2 + p^2$

2 → n

$p2 = p1$

$p1 = curr$

Array = O(n)

T/C = O(n)

S/C = O(1)

Array → ✗

Back Tracking

Tries

2hrs [Stacks] Heaps

Adv Graph Algos

P-1

9205

- Arrays
- Linked Lists
- Stacks
- Queues
- Trees
- BST
- Priority Queues
- Graphs
- Greedy
- DP
- Recursion
- Hashmaps