DP 3 cont Jan
Monday, 30 January 2023 9:06 PM

Q Given some coins & their values
Given → N
In how many ways you can make N using the coins.
→ Each coin is present an terms!

N=5, A: [1, 2, 3]

1) Order doesn't matter 2) Order matter ([1+1+1], [1+1+2], [1+2+1])

1+1+1+1+1 → 1
1+1+3 → 1
1+4 → 2
} → 6

Q 0-1 order matter

dp(N) → no. of ways to making N sum using the given arr

$$dp(i) = \sum_{j=1}^{|A|} dp[i-A_j]$$

Base Cases
dp(0) → 1

dp(i) = dp(i-1) + dp(i-2) + dp(i-3)

```
int ways(i) {
    if (i==0) ret 1;
    if (dp[i] != -1) ret dp[i];
    ans=0;
    for (j:0 → |A|-1) {
        if (i-A_j ≥ 0)
            ans += ways(i-A_j);
    }
    dp[i] = ans;
    ret ans;
}
```

→ dp[N+1] = -1 (0)
→ dp[0] → 1

```
for (i: 1 → N) {
    for (j: 0 → |A|-1)
        if (i-A_j ≥ 0)
            ans += dp[i-A_j];
    dp[i] = ans;
}
```

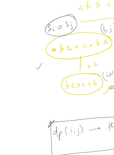
→ if (i-A_j ≥ 0)

N=6

1+1+1+1+1+1
1+1+1+3
1+1+4
3+3
1+5
3+2

→ SS → O(N×|A|)
O(N)

TC: O(N×|A|)



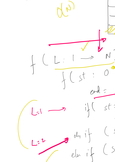
dp(N, idx)
→ dp(N, idx-1)
→ dp(N-A_idx, idx)

(idx, N)
(idx-1, N)
(idx, N-A_idx)

SC: O(N×|A|)
O(N)

Q Given a string.
Find the length of the longest Palindromic Subsequence.

a b c d e c a b x b



S[i:j] → abcxcba
(i,j)
(i+1, j-1)
(i, j-1)
(i+1, j)

R[0:i]

dp(i,j) → the LPS from start to the string S[i:j]

S[i]==S[j] → 2 + dp(i+1, j-1)
dp(i,j)
max → dp(i+1, j)
→ dp(i, j-1)

Base Cases
i==j → 1
i>j → 0

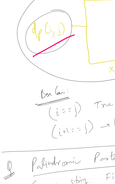
SS → N²
TRFS → O(1)
TC: O(N²)
SC: O(N²)

TC: O(N²)
SC: O(N²)

dp(0,N)

f(L: 1 → N) {
    f(st: 0 → N-1) {
        end: st+L-1
        if (st==end)
            dp[st,end] = 1
        else if (st+1==end)
            dp[st,end] = (S[st]==S[end]) ? 2 : 1
        else if (S[st]==S[end])
            dp[st,end] = 2 + dp[st+1, end-1];
        else
            dp[st,end] = max(dp[st+1, end], dp[st, end-1]);
    }
}

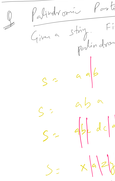
L=1 → if
L=2 → else if



Q Given a string
If substring check if it is palindrome or not.

S → N
SS → N²
TC: O(N³)
check → O(N) / S.S

dp[i,j] → if the substring S[i:j] is a palindrome or NOT!



Base
S[i]==S[j]
dp[i+1, j-1]
dp[i,j]
True
False

Base Cases
(i==j) True
(i+1==j) → (S[i]==S[j])

TC: O(N³)
N²
f(i:0 → N-1)
f(j: i → N-1)
check(i,j)

Q Palindromic Partitioning
Given a string. Find the min cuts to make palindromic partitioning.

S: a a b → 1
S: a b a → 0
S: a b b a c c → 4
S: x y z z y x → 4

dp(i) → min no of cuts req. S[0:i]

$$dp(i) = \min_{0 \le j \le i} \{ 1 + dp(j) \}$$
SS S[j+1:i] is palindrome

N

SS → N
TRFS → O(N × N)
check → O(N²)
O(N²)
TC: O(N³)

Opt
1) Solve for prev prob. O(N²)
* SS check if is → O(1)

2) Solve this problem X using the same DP to check if in SS is palindrome → TRFS → O(N)
SS → O(N)
TC: O(N²)

check

S: a b c b
0 1 2 3
(0,1)

	0	1	2	3
0	T	F	F	F
1		T	F	T
2			T	F
3				T