Q Given an array of non negative integers. Return True if there exist a non empty subset with a given sum K.

$$A = \begin{array}{c} \\ \\ \end{array}\begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ \end{array}$$

$$A = [3, 34, 3, \underline{4}, 12, \underline{5}, 2]$$

$K = 9$

$\{4, 5\}$

$\{3, 4, 2\}$

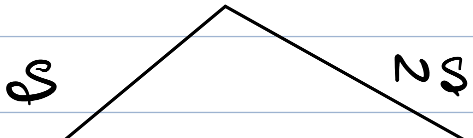
Sol^n 1) Brute Force

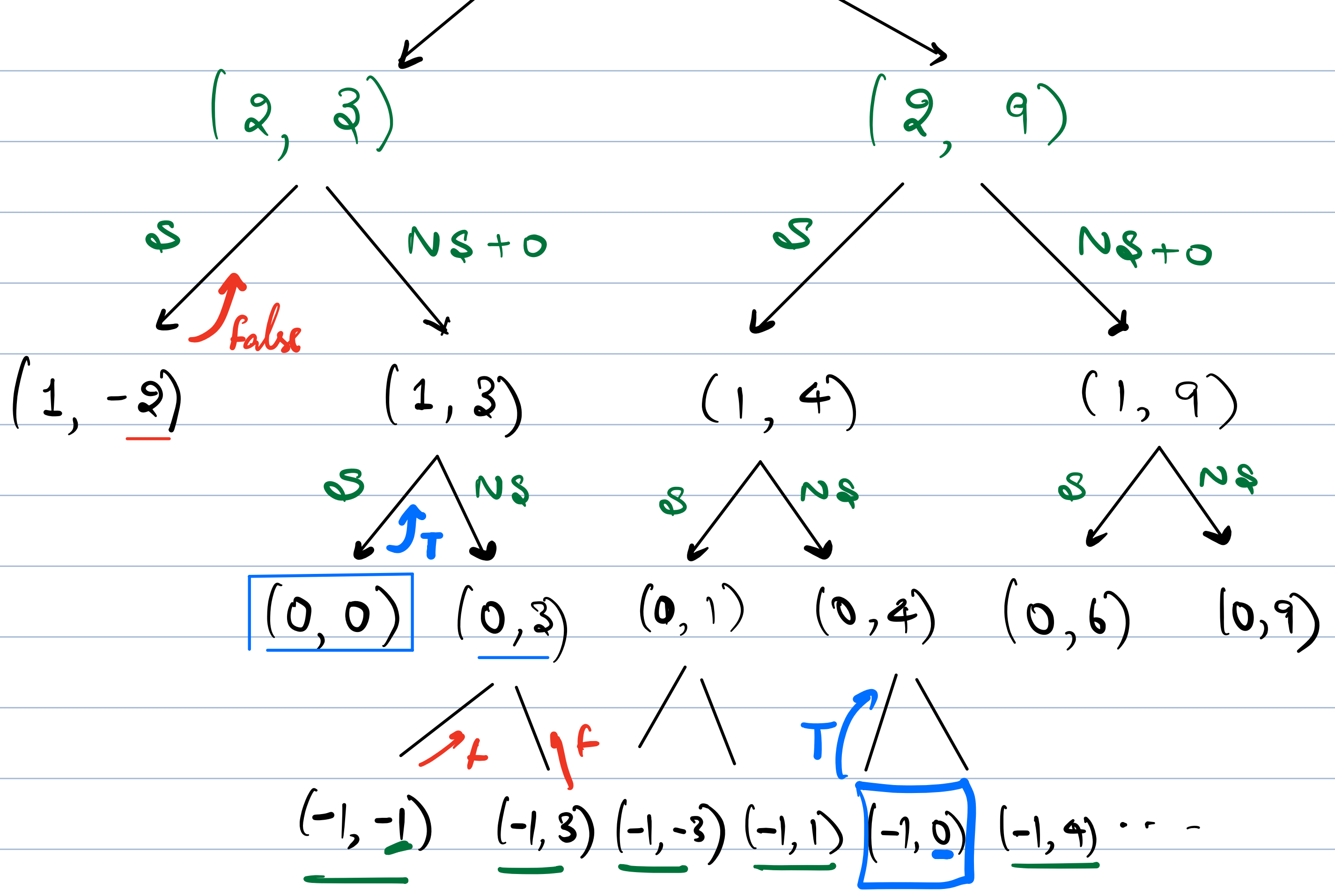
Generate all subsets & calculate Sum.

$$\begin{array}{cccc} 0 & 1 & 2 & 3 \\ \end{array}$$

$$A = [\underbrace{\underline{4}, 3, \underline{5}}, \underline{6}] \qquad \underline{K=9}$$

$(3, 9)$





Code

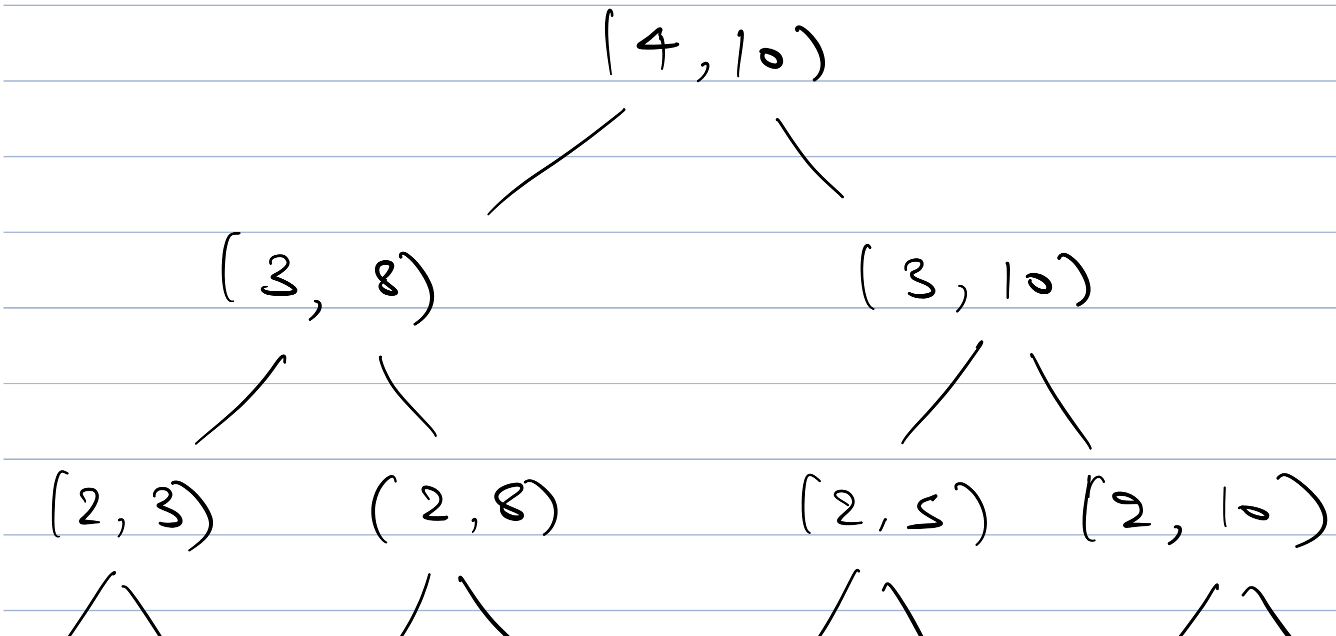
```
                          0/N-1       K
                            ↑          ↑
bool    checkSum ( index ,  remSum) {

    if (remSum == 0) {
                  return  True;
    }
    if (remSum <0) {
                  return  false;
    }
    if (index == -1) {
                  return  false;
    }

    return checkSum(index-1, remSum - A[index])
           || checksum(index-1, remSum);

}
```

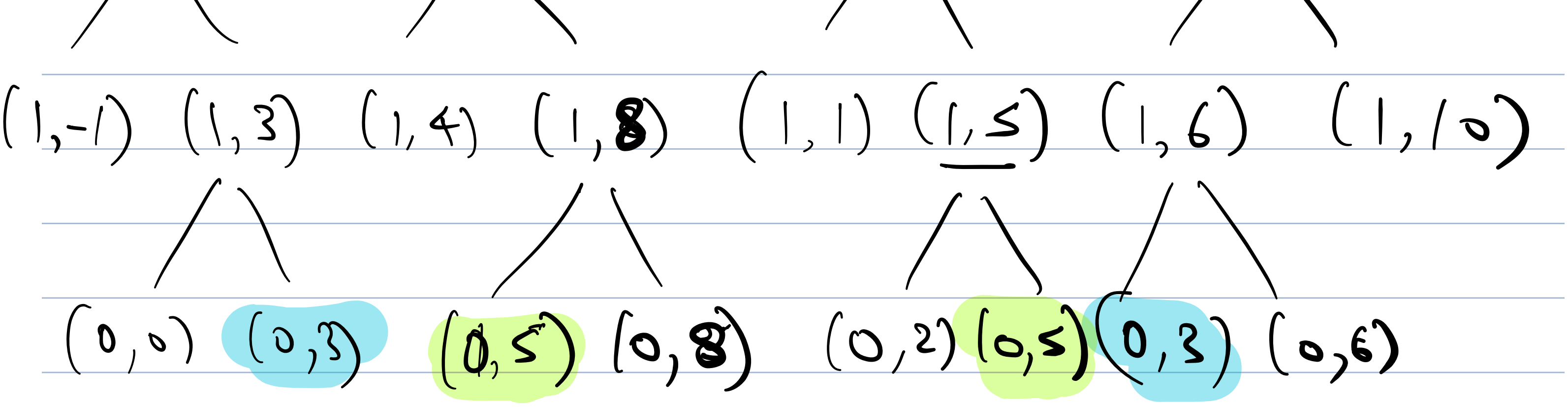
$T.C. = O(2^N)$

$S.C. = O(\underline{N})$

$K = 10$

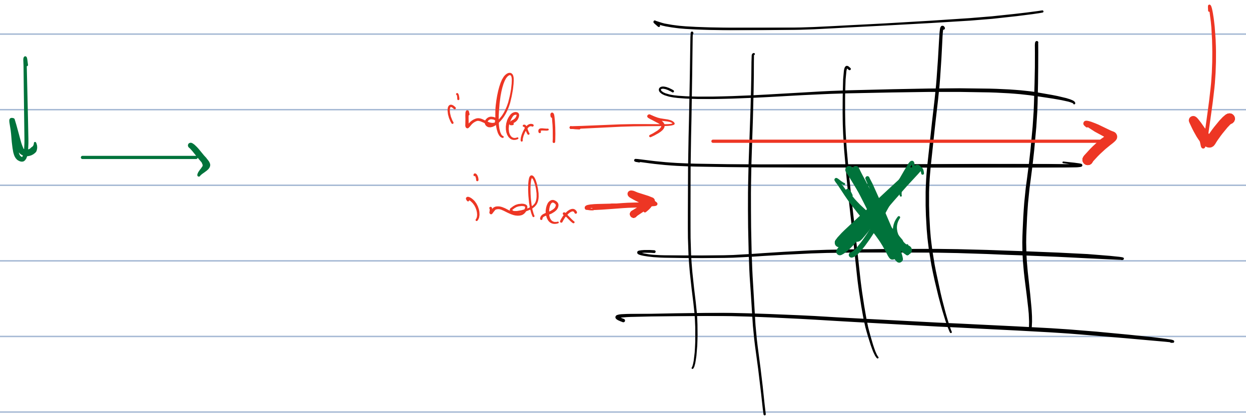
$$A = \begin{array}{c} \\ \end{array}\left[\begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ 1, & \underline{2}, & \underline{4}, & \underline{5}, & 2 \end{array}\right]$$





index ↑ remSum ↑

bool DP[N+1][K+1] = { false };



```
for(i=0;  i<=N;  i++){
              DP[i][0] = True;
}
```

```
for (i=1;  i<=N;  i++){          /N
        for (j=1;  j<=K;  j++){      /K
            if (A[i-1] <= j){
                    DP[i][j] = DP[i-1][j-A[i-1]]
```

```
                              || DP[i-1][j];
            } else {
                  DP[i][j] = DP[i-1][j];
            }
      }
}

return DP[N][K];
```

$$T.C. = O(N \times K)$$

$$S.C. = O(N \times K)$$

$$\downarrow$$

$$\underline{\underline{O(2 \times K)}}$$

Q// Flipkart ⇒ Offer ⇒ Recommend

C → Amount of money

Recommend the best products (Ratings)

⇓

Happiness

Product → Price
→ Happiness / Rating

Goal: Maximum Happiness value you can give any person.

⇓

(0 - 1 KnapSack)

Q Given an array of size N.

A[i] represents the max. size of jump that you can make from the index i

Min no. of jumps required to reach end.

Initially present at index 0

$$A = \left[\begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ 2, & 3, & 1, & 1, & 4 \end{array}\right]$$

2 jumps

Sol^n

1) Elements of Choice

$[1, A[i]]$

2) State

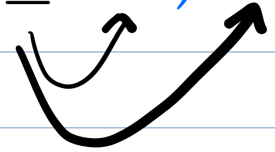
$\text{minJumps}(i) \Rightarrow$ Min no. of jumps required to reach $(N-1)$ from i.

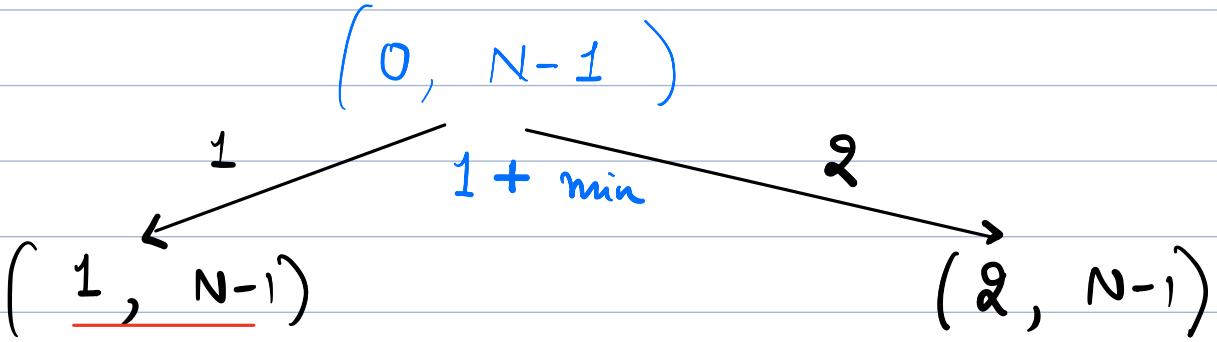
3) Recurrance rel^n

$$\text{minJumps}(i) = 1 + \min_{\forall j \text{ from } 1 \text{ to } A[i]} \left(\text{minJumps}(i+j) \right) \quad \text{if } (i+j < N)$$

$$A = \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ \hline 2, & 3, & 1, & 1, & 4 \end{array}$$

2 jumps





4) State that contains the final answer ??

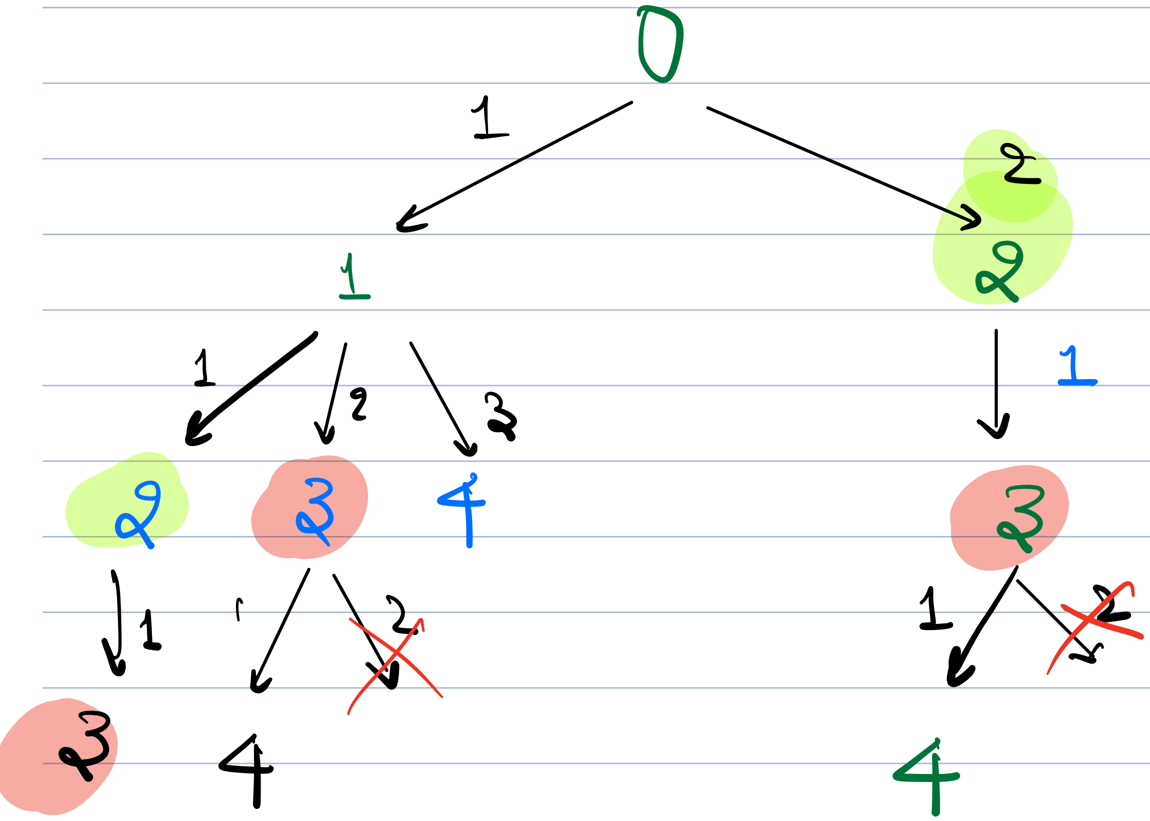
minjumps(0)

Code

```
int minJumps( $\overset{0}{i}$ ) {
      if (i == N-1) {
                  return 0;
      }

   ans = INT_MAX;
   for (j = 1; j <= A[i]; j++) {
      if (i + j < N-1) {
            ans = min (ans, 1 + minJumps(i+j));
      } else {
            break;
      }
   }
   return ans;
}
```

! Overlapping Subproblems.

$$A: \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ 2, & \underline{3}, & 1, & 2, & 4 \end{array}$$



DP

Code

```
DP[N] = {-1}

int minJumps(i) {

    if (i == N-1) {
        return 0;
    }
    if (DP[i] != -1) {
        return DP[i];
    }
  ans = INT_MAX;
  for (j = 1; j <= A[i]; j++) {
     if (i + j <= N-1) {
         ans = min (ans, 1 + minJumps(i+j));
     } else {
         break;
     }
  }
  DP[i] = ans;
  return ans;
}
```

$(i+j) \leq \min(N-1, i+A[i])$

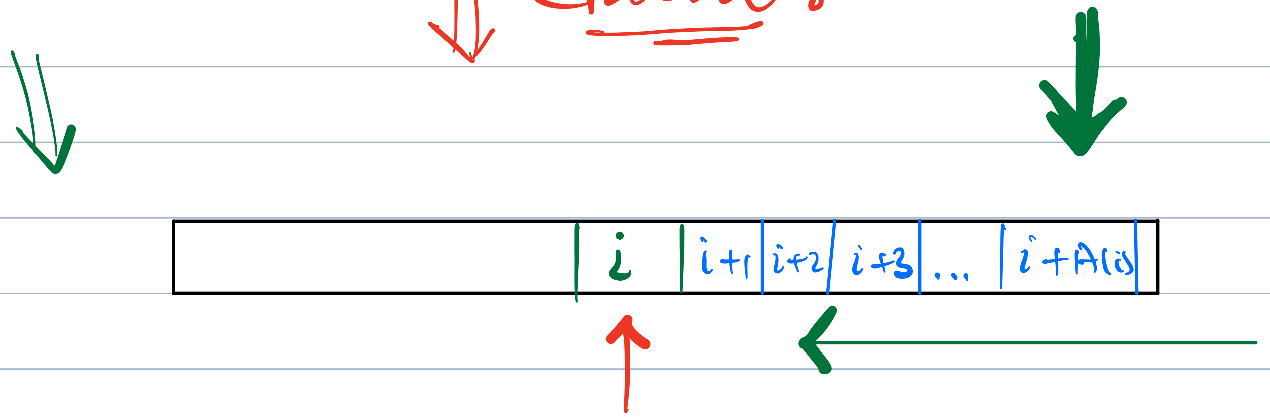
↑+−

$$T.C. = \left(\#\text{Recursive States}\right) \times \left(T/RS\right)$$

$$N \times N$$

$= O(N^2)$

$\Downarrow$ Tabular

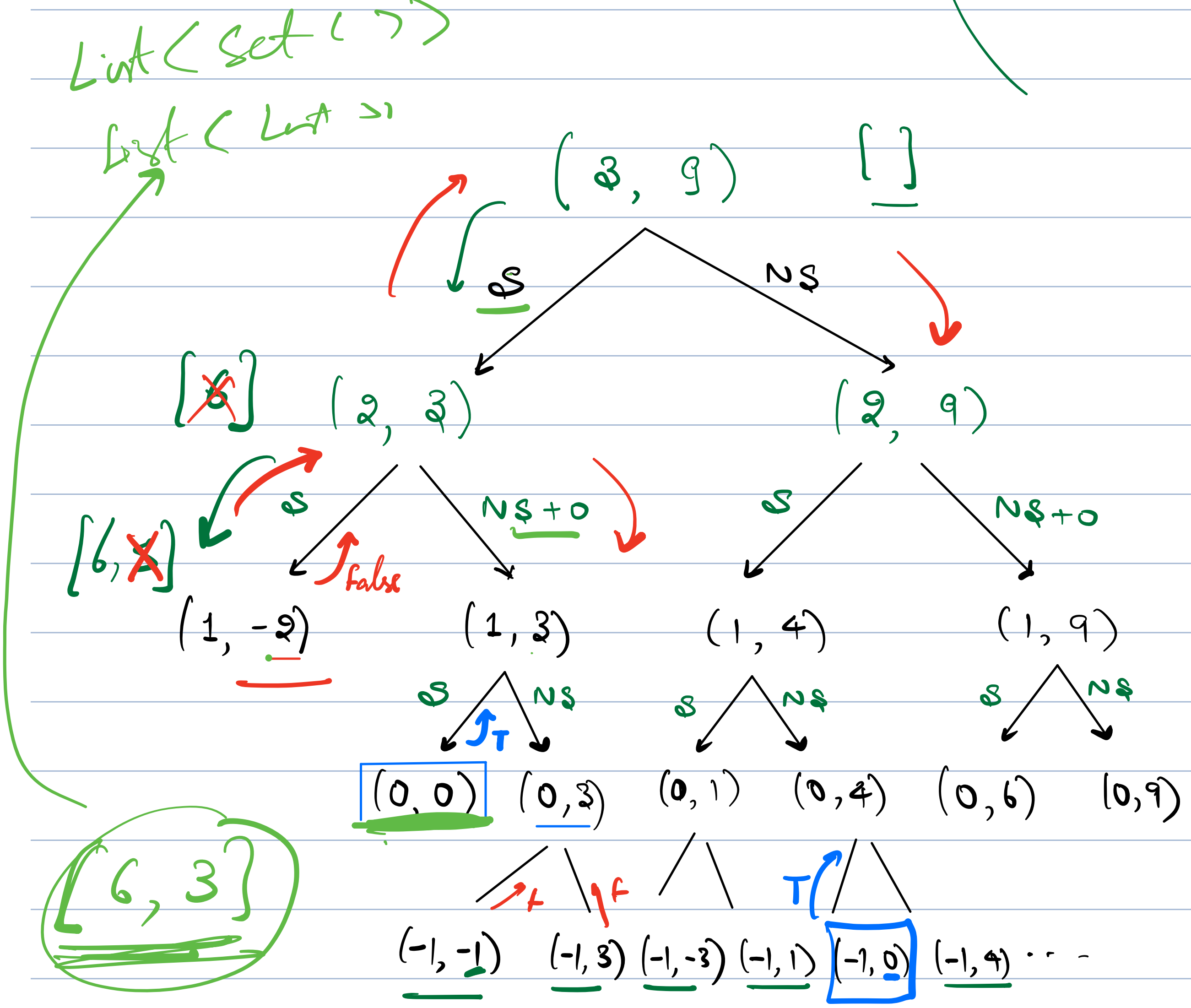


```
DP[N-1] = 0;

for (i = N-2; i >= 0; i--) {
        ans = INT_MAX;
        for (j = 1; (i+j) <= min(N-1, i+A[i]); j++) {
                ans = min(ans, 1+DP[i+j]);
        }
        DP[i] = ans;
}

return DP[0];
```

$T.C. = O(N^2)$

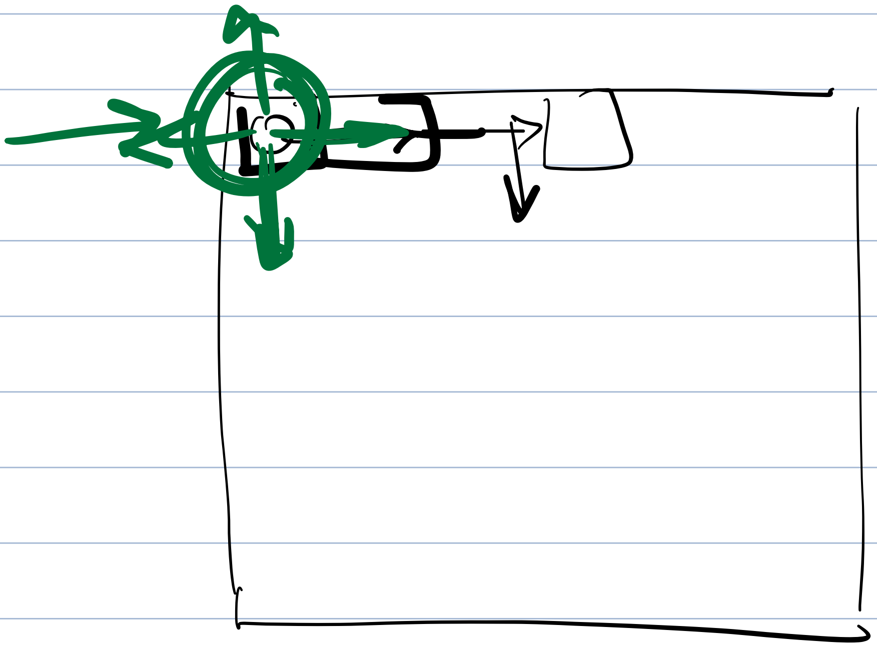
$S.C. = O(N)$

Best Time to Buy & Sell Stock ⇒ (Max Sum w/o adjacent elements)



Contest ⇒

$\underline{N^2}$



$N^2 - 1$