1463. Cherry Pickup II

→ fixed starting point.

$\boxed{2}$	3	1	$\boxed{2}$
0	$\boxed{4}$	2	$\boxed{2}$
5	$\boxed{8}$	3	$\boxed{5}$

$12 + 9 = 21$

→ Will Greedy work :- Greedy will not work here bcz of uniformity nature bcz in the grid cherries are not uniformaly distrubuted & griedy works on uniform distribution.

→ Recursion :-

(All path by robo1 + All path by robo2) & take maximum of them.

IMP :- In order to write recurrance we have to make sure that robo1 & robo2 move together bcz it might happen there is common cell in robo1's path & robo2's path & that consider only once.

You don't be like that let's do for robo1 first & get the maximum sum & then for robo2 in that case you have to traverse all the path for both & if their are something common then you have to subtract that cell from actual answer & this will be much longer.

Rules for Recursion

(1) express every thing in terms of (i1, j1) & (i2, j2) and write down all the base cases

(2) explore all the paths

(3) return the maximum sum

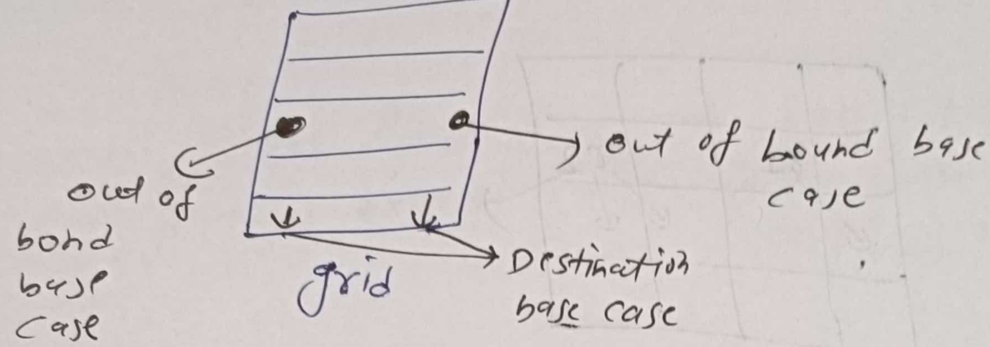
Whenever you have fixed starting points & variable ending points then start recursion from fixed starting point.

Rule-1 write down the Base cases

There are two types of Base case

Destination Base case

out of Bound Base case.



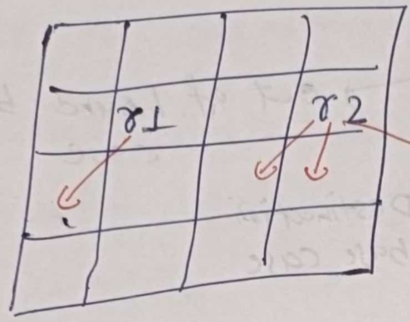
If any robo go out of bound then that path should not be considered.

f(i, j1, j2)
Here we are only taking i not i1 & i2 bcz for both robo i1 & i2 will be same.

```
f(i, j1, j2)
{
    // out of Bound Base case
    if(j1<0 || j1>=m || j2<0 || j2>m)
        return -1e8;
    // Destination Base case.
    if(i==n-1)
    {  if(j1==j2) return a[i][j1];
       return a[i][j1] + a[i][j2];
    }

    // Explore all possibilites.
       maxi=0
    for(dj1 -> -1 to +1)
    {  for(dj2 -> -1 to +1)
       {  if(j1==j2)
          {  maxi= max(maxi, f(i+1, j1+dj1, j2+dj2) + arr[i][j1]
          }
          else
          maxi= max(maxi, a[i][j1] + a[i][j2] +
                         f(i+1, j1+dj1, j2+dj2)).
       }
    }
    return maxi;
```

③ explore all possibilites:



i.e. if robo1 → r1 move left then robo2 → r2 can move in three different cell

so total movements are = 9

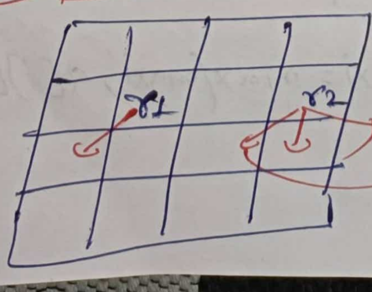
so there are 9 possibilities of path.

$$\boxed{\begin{aligned} T.C. &= O(3^n * 3^n) \\ S.C. &= O(n) \leftarrow \text{stack space} \end{aligned}}$$

→ DP Memoization

$$\boxed{\begin{aligned} T.C. &= O(N*M*M) * 9 \\ S.C. &= O(N*M*M) + O(\tilde{n}) \end{aligned}}$$

why DP



here r2 is already explore this & might possible r1 comes here.

Tabulation

Rules
- (i) first write the base cases.
- (ii) express every state in for loop
  - states → i, j1, j2.

(i) Base case

$dp[n][m][m]$

// first Base case → Destination Base case.
- if (i == n-1) → for i = n-1 the j1 → 0 to m-1, j2 → 0 to m-1

```
for(j1 -> 0 to m-1)
{ for(j2 -> 0 to m-1)
  { if( j1 == j2) dp[n-1][j1][j2] = grid[n-1][j1];
    else dp[n-1][j1][j2] = grid[n-1][j1] +
                            grid[n-1][j2];
  }
}
```

// for loop

```
for(i=n-2 to 0)
{ for(j1= 0 to m-1)
  { for(j2= 0 to m-1)
    {
      ==== <- copy the recurrence
    }
  }
}
```