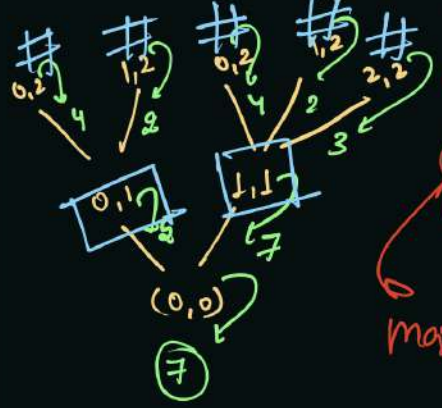
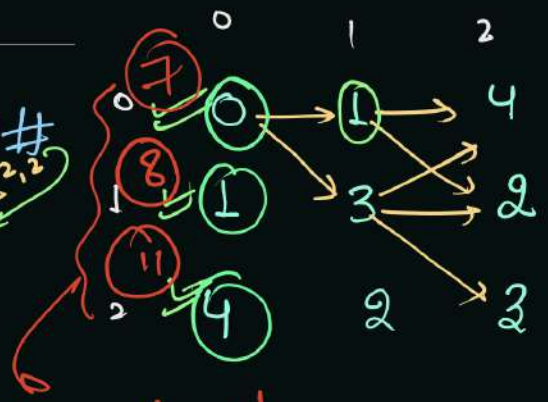
Goldmine

Thursday, 24 June 2021 6:13 PM

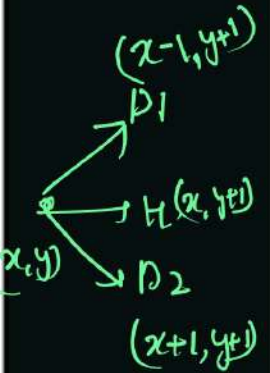




main function

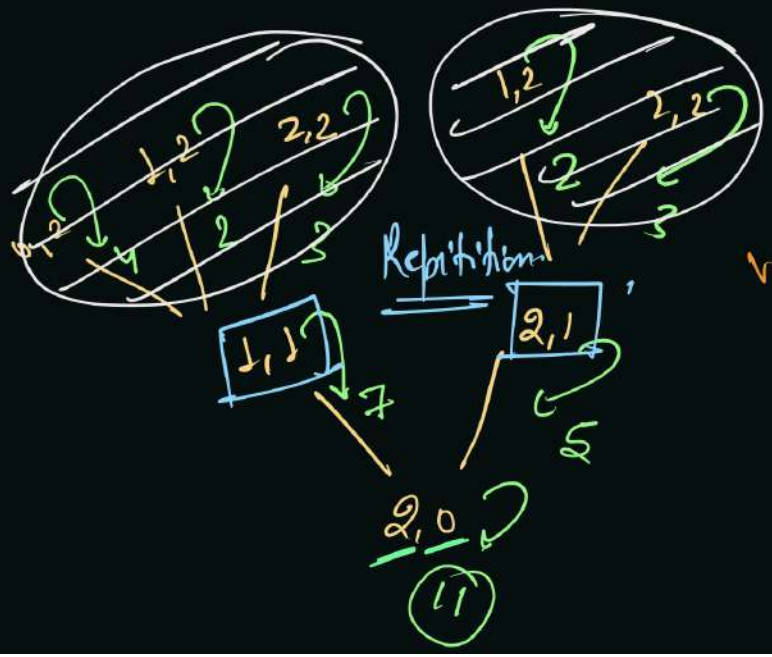
helper function (Rec)

	0	1	2	3	4	5
0	0	1	4	2	8	2
1	4	3	6	5	0	4
2	1	2	4	1	4	6
3	2	0	7	3	2	2
4	3	1	5	9	2	4
5	2	7	0	8	5	1



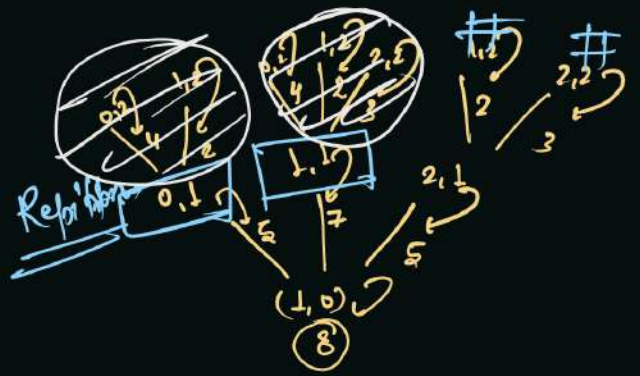
→ Base case

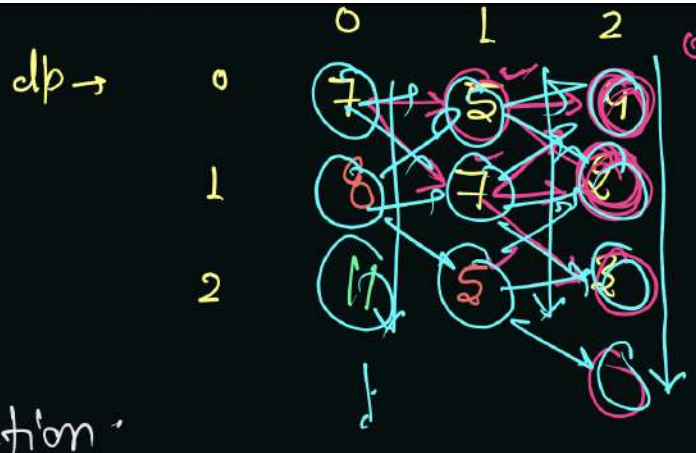
□ → Repitition



variables of Repition → (x, y)

↳ 2D - DP





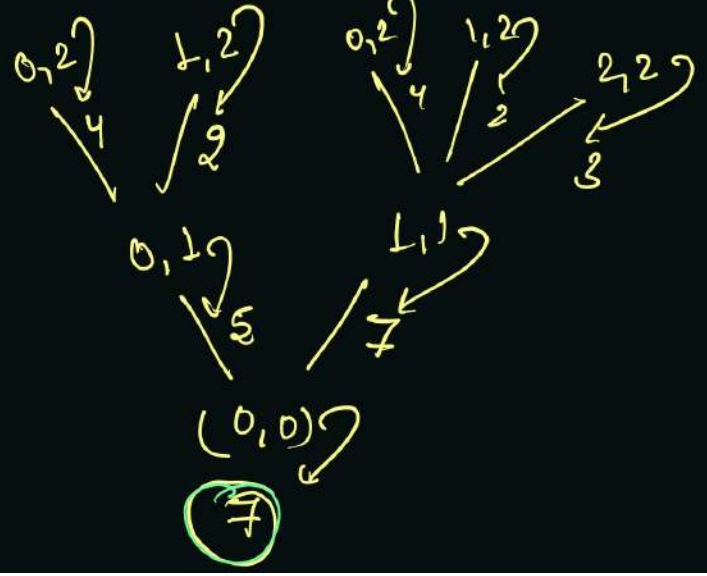
Order of Solving in Index

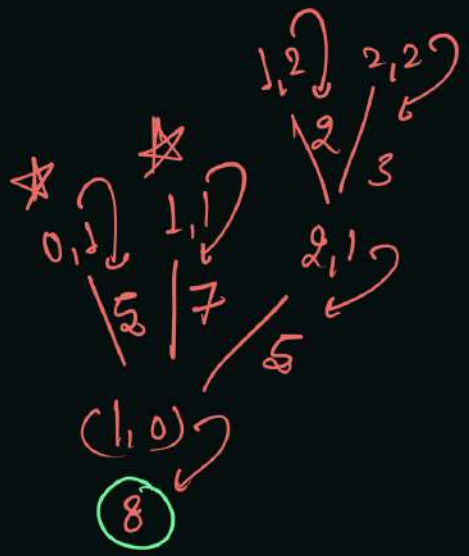
0,2
1,2
0,1
2,2
1,1
(0,0)

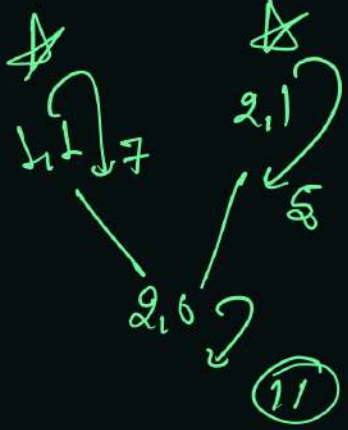
	0	1	2
0	0	1	4
1	1	3	2
2	4	2	3

Memoisation.

Final Result = max (7, 8, 11) = 11 Ans







Goldmine - Tabulation →

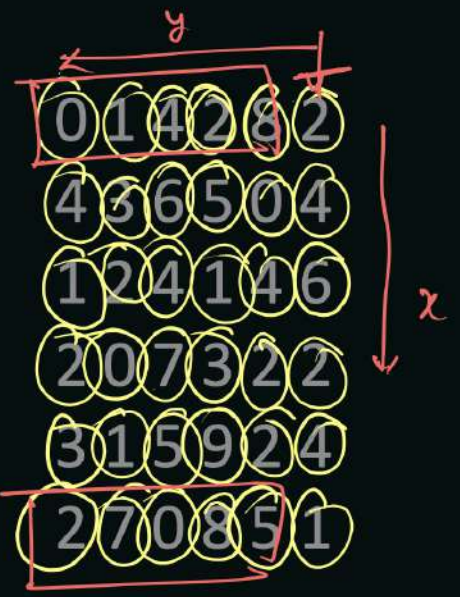
outerloop → Column
↳ Innerloop → for Row

dp →

	0	1	2	3	4	5
0	26	24	21	14	12	2
1	31	26	23	17	6	4
2	28	27	21	11	10	6
3	29	25	25	13	8	2
4	33	26	23	18	6	4
5	32	30	18	17	9	1

max from 0th Row is final result

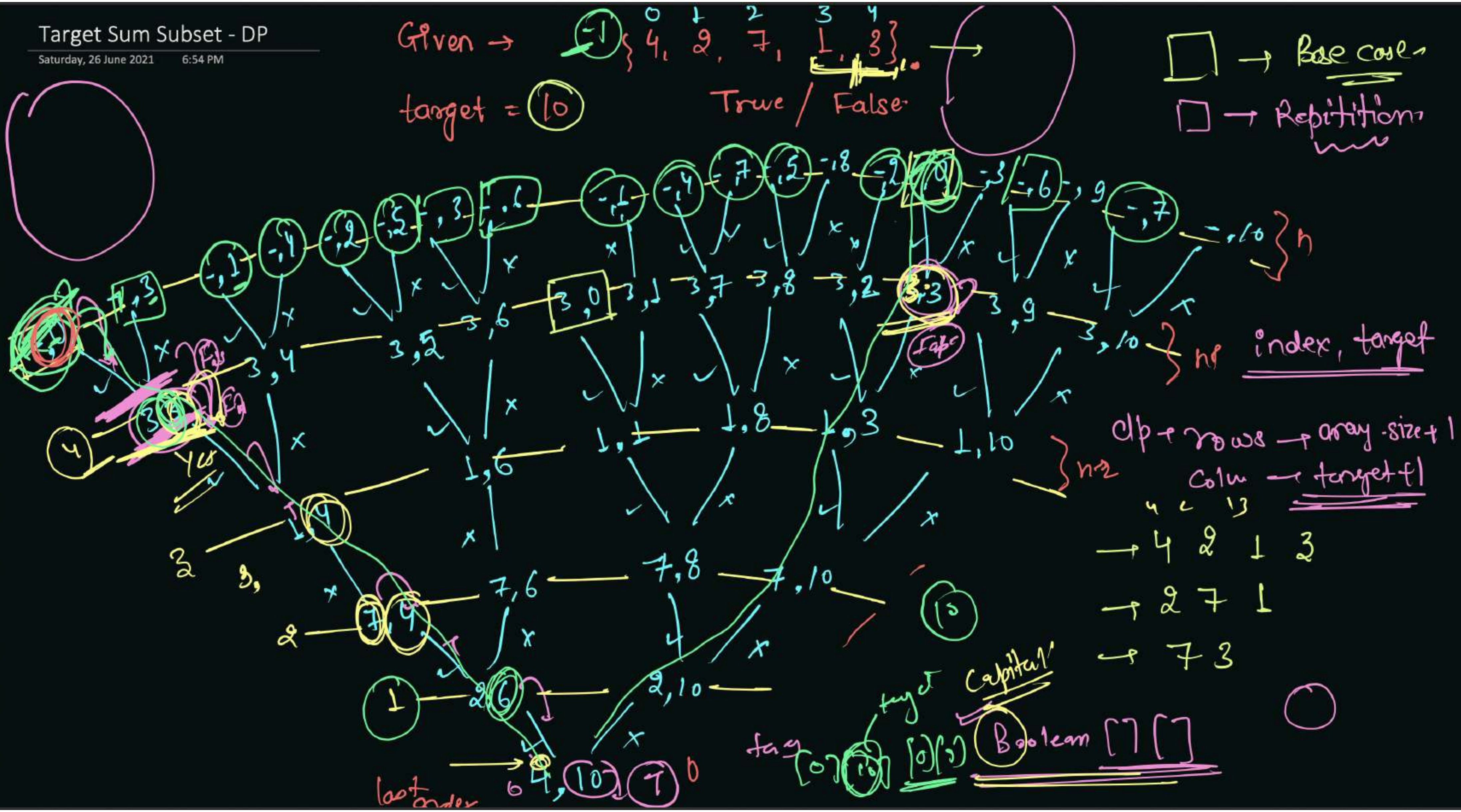
mine =



Final Result →

Target Sum Subset - DP

Saturday, 26 June 2021 6:54 PM



Target Sum subset →

{4, 2, 7, 1, 3} (indices 0 1 2 3 4), target = 10

dp[r][c] = dp[r-1][c] || dp[r-1][c-val];

(No call: dp[r-1][c]; Yes call: dp[r-1][c-val])

outerloop: target →
inner loop: indx ↓

array ↓ target →

index → value	0	1	2	3	4	5	6	7	8	9	10
0 → -	T	F	F	F	[illegible]	F	F	F	F	F	F
1 → 4	T	F	F	F	T	F	F	F	F	F	F
2 → 2	T	F	T	F	T	F	T	F	F	F	F
3 → 7	T	F	T	F	T	F	T	T	F	T	F
4 → 1	T	T	T	T	T	T	T	T	T	T	T
5 → 3	T	T	T	T	T	T	T	T	T	T	T

* Meaning of Cell

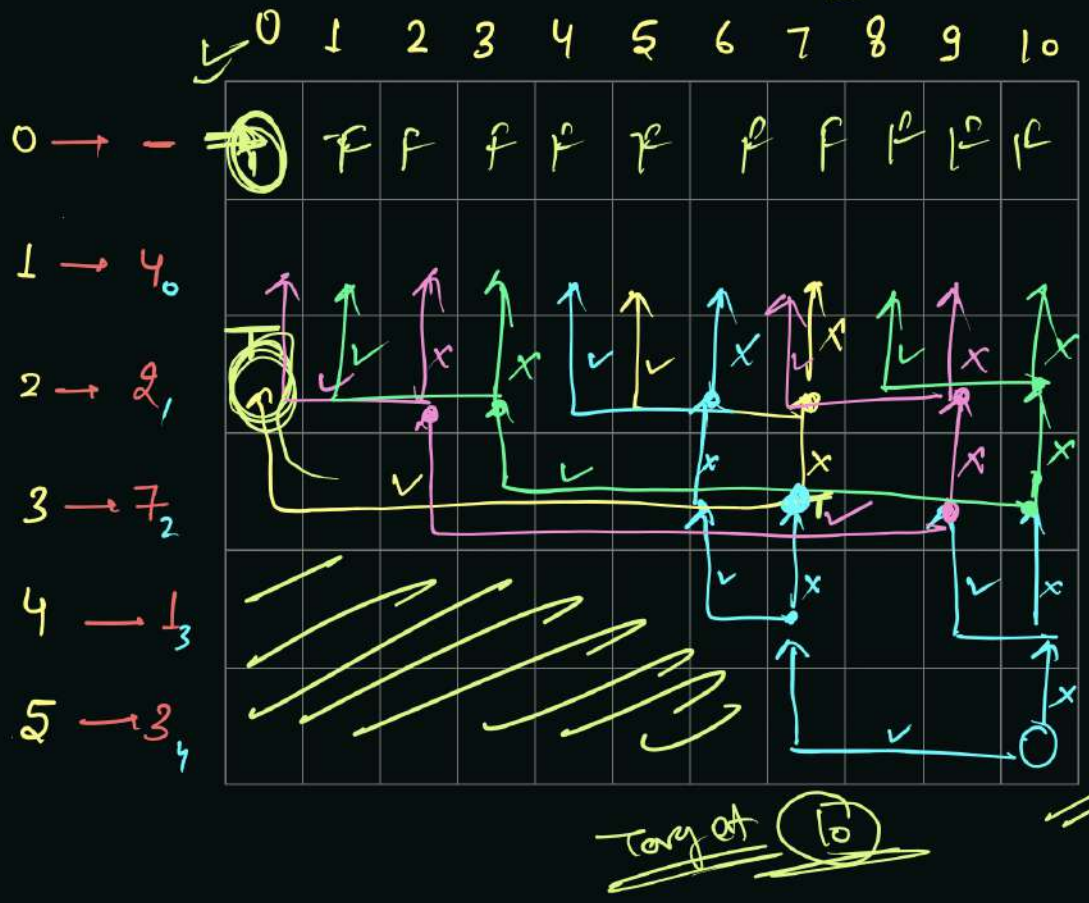
dp[i][j] can we make j target using elements from 0 to i

result = dp [5][10] result

target = 10

target = 0

Memoisation



Target 10

```java
public static boolean targetSumSubset_memo(int[] arr, int indx, int target,
    if(target == 0) return dp[indx][target] = true;
    if(indx == arr.length) {
        return dp[indx][target] = false;
    }

    if(dp[indx][target] != null) {
        return dp[indx][target];
    }

    boolean res = false;
    // yes call
    if(target - arr[indx] >= 0) {
        res = targetSumSubset_rec(arr, indx + 1, target - arr[indx]);
    }
    // no call
    res = res || targetSumSubset_rec(arr, indx + 1, target);
    return dp[indx][target] = res;
}
```

Tabulation

Memoisation

Better

arr, n-1, target

arr, 4, 10