

## (A) Subset sum

\* Using top down

Problem:- given an array of integers we need to find subset whose sum equals given sum

$a[] = \{2, 3, 8\}$  Sum = 11

Approach  $\rightarrow$  so while solving

[Sumset  
Matrix]

①  $\rightarrow$  sum should be less than element of array

② maintain a <sup>bool</sup> 2d DP array.

③ ~~for~~ columns = sum + 1  
rows = size + 1

④ Now as we know an empty  $\{ \}$  is <sup>always a</sup> subset so for  $i=0, j$  sum = 0 so initialize first col = 1

⑤ init an array of size 0 cent give any sum so initialize row as false/0

⑥ Now ~~check~~ if element is  $>$  sum at given instant  $dp[i][j] = dp[i-1][j]$

i.e copy the value ~~is~~ is just above row

else ⑦  $dp[i][j]$  ~~is~~  $no$  = we will take either of element value ~~is~~ given by element before or we will go to the position of diff ~~is~~ sum - give element and take its value.



Use  $arr = [2, 3, 8]$   $sum = 11 \rightarrow$  empty string

col = 11 + 1 row = 3 + 1

sum size	0	1	2	3	4	5	6	7	8	9	10	11
0	T	F	F	F	F	F	F	F	F	F	F	F
1	T	F	T	F	F	F	F	F	F	F	F	F
2	T	F	T	T	F	T	F	F	T	F	T	T
3	T	F	F	T	F	T	F	F	T	F	T	T

\* consider size of array of size [1] with last element for  $sum < 2 = \text{False}$   
 $\rightarrow$  Now we have choice either 2 take or 2 leave

\* cons size = 2  $sum = \{2, 3\}$  concept of choice  
 Now for each value of sum  
 make choice  $\rightarrow$  3 take or  $\rightarrow$  3 leave

\* size = 3  $sum = \{2, 3, 8\}$   
 make choice  $\rightarrow$  8 take or  $\rightarrow$  8 leave

for  $sum < 8 = \text{copy value above its row}$   
 because it may be completed by previous set

we will return last element



code  $\rightarrow$  if (sum > arr[i-1]) ~~return~~  
 $dp[i][j] = dp[i-1][j]$ ;

else  
 $dp[i][j] = dp[i-1][j]$  or  
 ~~$dp[i-1][j-arr[i]]$~~   
 $dp[i-1][j-arr[i]]$   
 when we don't take arr[i]      when we take arr[i]

memoization

memoization  $\rightarrow$  maintained 2d  $dp[i][j]$   
 $\rightarrow$  and set all elements to -1

pseudo code  $\rightarrow$

if (sum == 0) return true;  
 if (n == 0) return false;

// if element is already in dp  
 if ( $dp[n-1][j] \neq -1$ )  
 $\{$  return  $dp[n-1][j]$   
 $\}$

~~if (sum < arr[n-1])~~  
 // if element greater than sum  
 // go to next element  
 if (sum < arr[n-1])  
 $dp[n-1][sum] = dp[n-2][sum]$

// else take the element or exclude it  
 $dp[n-1][sum] = fn(arr, n-1, sum)$  or  
 $fn(arr, n-1, sum - arr[n-1])$