SUBSET SUM

Question :

Given an integer array nums, return true if you can partition the array into two subsets, such that the sum of the elements in both subsets is equal or False otherwise.

EXAMPLE :

Input : nums = [1,5,11,5]
Output : true
Explanation : The array can be partitioned as [1,5,5] and [11].

Constraints :

$1 \leq$ nums.length ≤ 200
$1 \leq$ nums[i] ≤ 100

CODE

```
VECTOR<INT> NUMS;
INT DP[205][20005];

BOOL FUNC (INT i, INT SUM)
{
    IF (SUM == 0)  RETURN 1;
    IF (i < 0)  RETURN 0;
```

```
    IF (DP[i][sum] != -1)  RETURN DP[i][sum];

    -> Not consider ith index
    INT IsPOSSIBLE = FUNC(i-1, sum);
    -> Consider ith index
    IF (sum - NUMS[i] ≥ 0) IsPOSSIBLE |=
            FUNC(i-1, sum - NUMS[i]);
    RETURN DP[i][sum] = IsPOSSIBLE;
}

BOOL CANPARTITION()
{
    INT SUM = ACCUMULATE(NUMS.BEGIN(),
                         NUMS.END(), 0);
    IF (SUM & 1) RETURN FALSE; // ODD
    SUM = SUM / 2;
    RETURN FUNC(NUMS.SIZE() - 1, SUM);
}

INT MAIN()
{
    MEMSET(DP, -1, sizeof SIZEOF(DP));
    NUMS = {1, 5, 11, 5};
    COUT << CANPARTITION() << "\n";

    RETURN 0;
}
```

RECURSION TREE (For given input) W/? DP