Subsequence : Sequence generated by deleting 0 or more elements from your array.

	0	1	2	3	4	5	6	7	
arr[8] =	3	-2	0	1	8	7	4	9	
	×	✓	✓	×	✓	×	✓	×	{-2, 0, 8, 4}
	✓	✓	✓	✓	×	×	×	×	{3, -2, 0, 1}
	✓	✓	✓	✓	✓	✓	✓	✓	{3, -2, 0, 1, 8, 7, 4, 9}
	×	×	×	×	×	×	×	×	{ }
	×	×	✓	✓	✓	×	✓	✓	{0, 1, 8, 4, 9}

{1, 0, 8, 4, 9}

× subsequence

order matter.

Subset

[1, 2, 4] $\xrightarrow[\text{descending}]{\text{Sort in}}$ [4, 2, 1]

{ }	{ }
{1}	{1}
{2}	{2}
{4}	{4}
{1, 2}	{2, 1}
{2, 4}	{4, 2}
{1, 4}	{4, 1}
{1, 2, 4}	{4, 2, 1}

If we sort array

→ Subsequence will change

→ order in subs. will change

→ elements will be same

↳ max/min sum

[-1, 4, 3, 9]

	Subsequence	subarray
[-1, 4]	✓	✓
[4, 3, 9]	✓	✓
[4, 9]	✓	✗
[-1, 3, 9]	✓	✗
[9, 3]	✗	✗

1. Each subarray is a subsequence
2. Each subseq. is need not a subarray.

Subset → same as subsequence, but order doesn't matter.

4, -1, 2 $\xrightarrow{\text{Sort}}$ -1, 2, 4

{ }	{ }
{4}	{4}
{-1}	{-1}
{2}	{2}
{4, -1} order doesn't matter	{-1, 4}
{-1, 2}	{-1, 2}
{4, 2}	{2, 4}
{4, -1, 2}	{-1, 2, 4}

$\{4, -1, 2\} = \{-1, 4, 2\} = \{2, 4, -1\} \ldots$ ——

Q. Given array N. How many subsequence are there.

$a_1 \quad a_2 \quad a_3 \quad \ldots\ldots \quad a_N$

$0 \swarrow \searrow 1 \quad 0 \swarrow \searrow 1$

$2 \quad 2 \quad 2 \quad \ldots\ldots \quad 2 = 2^N$

$\{a_1\} = \{\}, \{a_1\}$ 2

$\{a_1, a_2\} = \{\}, \{a_1\} \{a_2\} \{a_1, a_2\}$ 4

$\{a_1, a_2, a_3\} = \{\}, \{a_1\} \{a_2\} \{a_3\} (a_1, a_2) (a_2, a_3) \{a_1, a_3\}$
$\{a_1, a_2, a_3\}$ 8

Q. Given Array N, how many subsets?

2^N

$\{a_1\} = \{\}, \{a_1\}$ 2

$\{a_1, a_2\} = \{\}, \{a_1\} \{a_2\} \{a_1, a_2\}$ 4

$\{a_1, a_2, a_3\} = \{\}, \{a_1\} \{a_2\} \{a_3\} (a_2, a_1) (a_2, a_3) \{a_1, a_3\}$
$\{a_1, a_2, a_3\}$ 8

Q. Given N distinct element,

Check if there exists a subset with sum = K

0	1	2	3	4	5	6

arr[7] = 3, -1, 0, 6, 2, -3, 5

K = 10

{-1, 6, 5}
{3, 2, 5}
{6, 2, -1, 3}

ans = True

K = 20

ans = False

Idea :

PF ✗

Sliding window ✗

Hashset ✗

2 loops ✗

Subarray ✗

Carry forward ✗

1, 3, 7, 19   K = 20

Check every subset, if sum is K or not.

arr[3] = {3, -2, 1} → 8 subsets ⟶ [0, 7]

→ 2^N subsets

0	=	0 0 0	{ }
1	=	0 0 1	{1}
2	=	0 1 0	{-2}
4	=	1 0 0	{3}
3	=	0 1 1	{-2, 1}
6	=	1 1 0	{3, -2}
5	=	1 0 1	{3, 1}
7	=	1 1 1	{3, -2, 1}

N = 4
0 1 2 3
[3, -1, 2, 8]   16 subsets   [0, 15]

	3 2 1 0	
i = 0	0 0 0 0	{ }
i = 1	0 0 0 1	{3}
i = 2	0 0 1 0	{-1}
3	0 0 1 1	{3, -1}
4	0 1 0 0	{2}
5	0 1 0 1	{3, 2}
6	0 1 1 0	{-1, 2}
7	0 1 1 1	{3, -1, 2}
8	1 0 0 0	{8}
⋮		
15	1 1 1 1	{3, -1, 2, 8}

Subsets.

N $\quad 2^N$ subsets $\quad [0, 2^N - 1]$

Bitmasking

```
boolean checksum ( int arr[], N, K )
{
        2^N subset      (0, 2^N - 1)

    for( i=0; i < 2^N ; i++)
    {
        // ith subset sum == K
        sum = 0
        for( j=0 ; j<N; j++)
        {
            if( checkbit (i,j) == true )
                  sum = sum + arr[j]
        }

        if( sum == k)  return True
    }

    return False
}
```

If jth bit in i is set or not

Bit manipulation 2

Break

10:13 10:18

TC : $O(2^N * N)$

SC : $O(1)$

Q. Given N elements, find sum of all subset sum

[3, 1, 4]

{ }	→	0
{3}	→	3
{1}	→	1
{4}	→	4
{3, 1}	→	4
{1, 4}	→	5
{3, 4}	→	7
{3, 1, 4}	→	8
		32

Idea1: for every subset get the sum, maintain total sum

```
int subsetsums( int arr[], N)
{
    totsum = 0                 2^N subset      (0, 2^N - 1)

    for( i=0; i < 2^N; i++)
    {
        // i^th subset   sum == K
        sum = 0
        for( j=0 ; j < N; j++)
        {
            if( checkbit (i, j) == true)
                sum = sum + arr[j]
        }
        totsum = totsum + sum
    }
    return totsum
}
```

if j^th bit in i is set or not

Bit manipulat

$TC: O(2^N \times N)$

$SC: O(1)$

Idea 2 : Contribution technique.

$n_0 \times arr[0] + n_1\ arr[1] + n_2\ arr[2] + \text{-------}$

Total subsets

$\underline{2}\ \underline{2}\ \underline{2}\ \underline{2}\ \underline{2} = 2^5$

1 2 3 4 5 = 32

0 1 0 1

Subset having 3rd element present

$\underline{2} \times \underline{2} \times \underline{1} \times \underline{2} \times \underline{2} = 2^4$

1 2 3 4 5 = 16

0 1 0 1 1

$\{ \underline{3} \quad \underline{\quad 2^4 \quad} \}$

If N elements are there.

0th index → 2^{N-1} subsets

1th index → 2^{N-1} subsets

⋮

N-1 index → 2^{N-1} subsets.

$$\text{Totsum} = 2^{N-1}\,arr[0] + 2^{N-1}\,arr[1] + 2^{N-1}\,arr[2]$$

$$= 2^{N-1}(\text{sum of array})$$

TC: O(N)

SC: O(1)

Q. Given N distinct Number

Calculate $\left(\dfrac{\text{sum of all subsets sum}}{2^N}\right)$

$$= \frac{2^{N-1}(\text{sum of array})}{2^N}$$

$$= \frac{\text{sum of array}}{2}$$

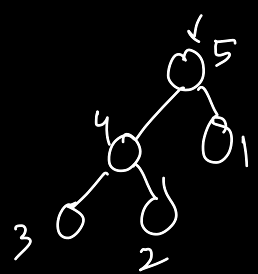
TC: O(N)

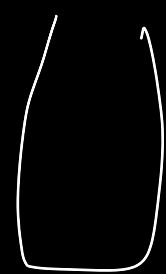
SC: O(1)

Adv DSA: Array - 1

Array - 2

Array - 3





Inorder(5)

{

Inorder(root.left)

root.data

Inorder(root.right)

}

Inorder(5)

Inorder(4)

Inorder(1)

Inorder(3)

Inorder(2)

3, 4, 2, 5, 1