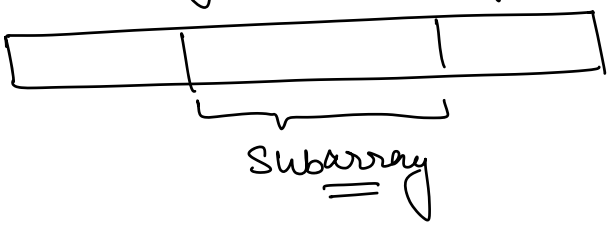
Subarray :- Contigous part of the Array.



Subsequence :- Sequence generated by deleting 0 or more elements from the Array

arr[8] : { -2 -3 6 2 4 -1 0 3 }

0 1 2 3 4 5 6 7

x x ✓ ✓ x x x ✓ {6, 2, 3} ✓

✓ ✓ ✓ x x x x ✓ {-2 -3 6 3} ✓

✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ {-3} ✓

x x x x x x x x { } ✓

x x ✓ x x x x x {6} ✓

{ 4 1 0 } Not valid subseq

⇒ In subsequences, Order of elements matters.

Ex :- A: { -1 4 3 9 }

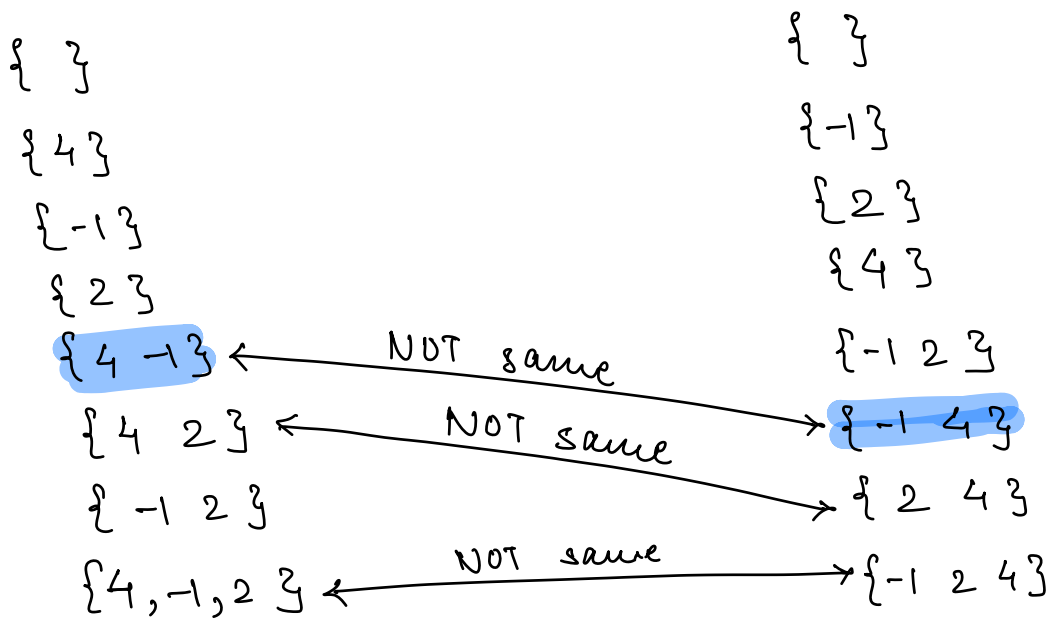
		Subsequence	Subarray
1)	{-1 4}	✓	✓
2)	{4 3 9}	✓	✓
3)	{4 9}	✓	X
4)	{-1 3 9}	✓	X
5)	{3 4}	X	X

⇒ Every subarray is a valid subsequence.

⇒ Every subsequence is NOT a valid subarray.

Ex :-

A: { 4 -1 2 } —SORT→ { -1 2 4 }



⇒ Subsequences of original array and the sorted array are NOT same.

SUBSETS

↳ Same as subsequence but order doesn't matter.

#

A: { 4 -1 2 } —SORT→ { -1 2 4 }

Subsets

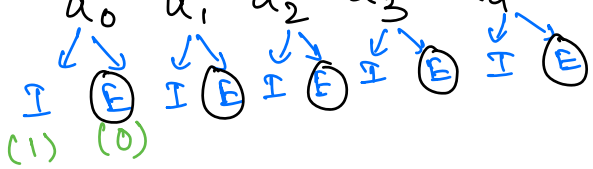
{ }
{4}
{-1}
{2}
{4 -1}
{4 2}
{-1 2}
{4 -1 2}

Same subsets

{ }
{-1}
{2}
{4}
{-1 2}
{-1 4}
{2 4}
{-1 2 4}

No. of subsequences / subsets ?

a[5] : a_0 a_1 a_2 a_3 a_4



$\Rightarrow 2 \times 2 \times 2 \times 2 \times 2 \Rightarrow 2^5$

for (N) elements :- 2^N

No. of subsequences excluding the empty :-

$2^N - 1$

↳ Empty subsequence

Q. Given N array elements, Check if there exists a subset with sum = K.

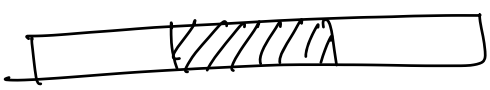
Arr : { 3 -1 0 6 2 -3 5 } K = 10

{ -1 6 5 }
{ 3 2 5 }
{ 3 -1 6 2 }
:
} True

No of subsets : 2^N

S.W / Prefix sum ? X

↳ Range sum



No of subsets : 2^N

{ -2 6 4 } (indices 0 1 2) # of subsets = 8

Bit masking

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

2 1 0 ← Indices

0 0 0
0 0 1
0 1 0 ←
0 1 1
1 0 0
1 0 1
1 1 0
1 1 1

Binary No's of N=3

{ } ⇒ 0
{ -2 } ⇒ -2
{ 6 } ⇒ 6
{ -2 6 } ⇒ 4
{ 4 } ⇒ 4
{ -2 4 } ⇒ 2
{ 6 4 } ⇒ 10
{ -2 6 4 } ⇒ 8

Brute force

```
for( i = 0; i < 2^N; i++ ){ ⇒ 2^n
    sum = 0;
    for( j = 0; j < N; j++ ){ ⇒ n
        if( checkBit(i, j) ){
            sum += a[j]
        }
    }
    if( sum == K)
        return true;
}
return false
```

1<<N

TC : O(N·2^N)

SC : O(1)

{ -2 6 4 } N=3 (K)

0 1 2

j ⇒ 0 to 7

i = 0 → 0 0 0 sum=0

j ⇒ 0, 1, 2

{ }

sum = 0

×

i = 2 ⇒ 0 1 0 ⇒ {6}

j → 0, 1, 2

× ✓ ×

sum += a[1]

sum = 6

i = 1 ⇒ 0 0 1 ⇒ {-2}

j ⇒ 0, 1, 2

↑ × ×

sum += a[0]

sum = -2

sum = -2

i = 3 ⇒ 0 1 1

j → 0, 1, 2

✓ ✓ ×

sum += a[0]

sum = -2

sum = 4

{-2, 6}

$O(N \cdot 2^N)$ → $O(2^N)$ → $O(N \cdot K)$

Brute force — Backtracking Sol^ — D.P

Q. Given an Array, find sum of all subset sums.

{ −2 (0) 6 (1) 4 (2) } # of subsets = 8

{ } ⇒ 0

{−2} ⇒ −2

{6} ⇒ 6

{−2 6} ⇒ 4

{4} ⇒ 4

{−2 4} ⇒ 2

{6 4} ⇒ 10

{−2 6 4} ⇒ 8

Sum = 32

⇒ −2×4 + 6×4 + 4×4

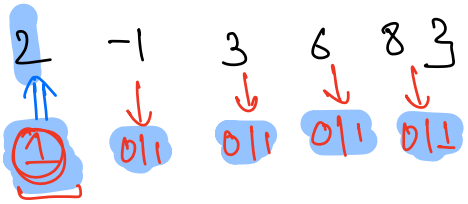
Contribution of (−2) in all subsets

Previous problem ⇒ Sol^ $O(N \cdot 2^N)$

Contribution Technique

↳ for every element, find in how many subsets it will appear.

arr[5] : { 2 -1 3 6 8 }

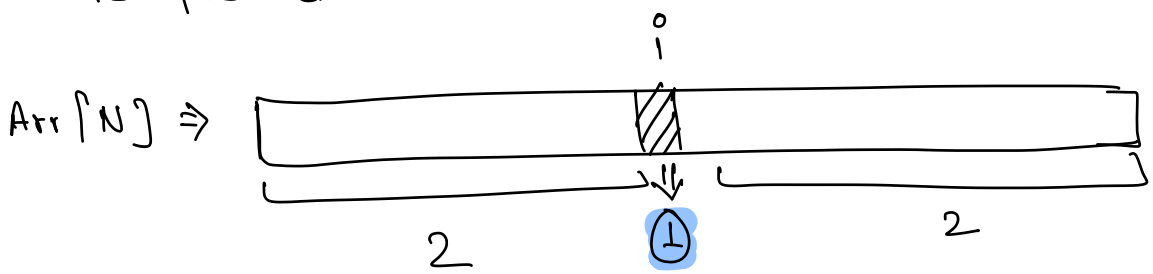


# of subsets in which (2) is present = $1 \times 2 \times 2 \times 2 \times 2 = 2^4$

{ 2 -1 3 6 8 }

2 (1) 2 2 2

# of subsets in which (-1) is present $\Rightarrow 2^4$



a[i] will be present in 2^{N-1} subsets.

Contribution of a[i] in sum of all subsets $\Rightarrow a[i] * 2^{N-1}$

Sum of all subset sums $= \sum_{i=0}^{N-1} a[i] * 2^{N-1}$

2^{N-1} → 1<<(n-1)

TC : O(N)
SC : O(1)

Q Given an Array, find the sum of all the subset sums divided by 2^N.

$$ans = \frac{\text{sum of all subset sums}}{2^N}$$

$$= \frac{\sum_{i=0}^{N-1} a[i] * 2^{N-1}}{2^N}$$

$$= \frac{2^{N-1} * \sum_{i=0}^{N-1} a[i]}{2^N}$$

$$ans = \frac{\sum_{i=0}^{N-1} a[i]}{2}$$

Q Given N array elements, calculate the sum of max of every subsequence.

facebook

{ 3 1 -4 }

	Max
{ }	0
{3}	3
{1}	1
{-4}	-4
{3 1}	3
{3 -4}	3
{1 -4}	1
{3 1 -4}	3
	10

{ -4 1 3 }

	Max
{ }	0
{-4}	-4
{1}	1
{3}	3 ✓
{-4 1}	1
{-4 3}	3 ✓
{1 3}	3 ✓
{-4 1 3}	3 ✓
	10

Sum = $0 \times 1 + -4 \times 1 + 1 \times 2 + 3 \times 4$

Sum = 10

$\Rightarrow$ { -2 8 0 4 3 }

sort ↓

{ -2 0 3 4 8 }

$\Rightarrow$ { -2 0 3 4 8 }

indices: 0 1 2 3 4

(2) (2) (2) (2) (1)

No. of subsequences in which (8) will appear as as MAX element.

$= 2 \times 2 \times 2 \times 2 \times 1$

$= 2^4$

{ -2 0 3 4 8 }

indices: 0 1 2 3 4

↓ ↓ ↓ ↓ ↓ (Exclusion)

2 2 2 1 1

No. of subsequences in which (4) will appear as as MAX element. $= 2 \times 2 \times 2 \times 1 \times 1 = 2^3$

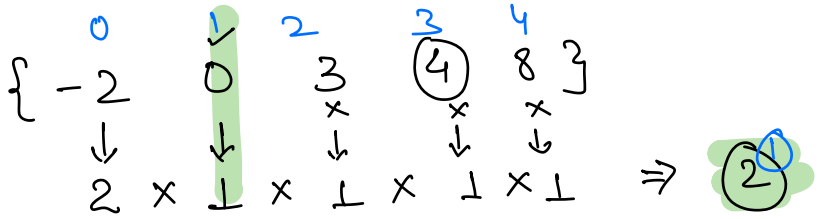
{ -2 0 3 4 8 }

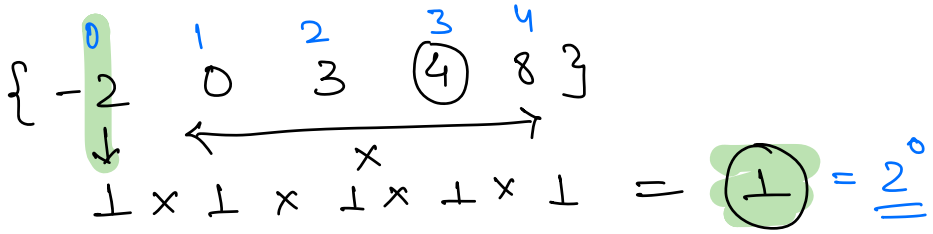
indices: 0 1 2 3 4

↓ ↓ ↓ ↓

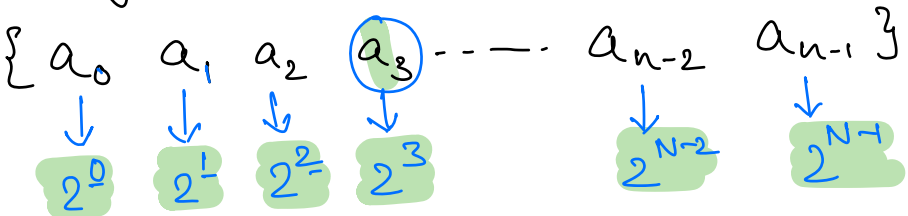
✓ × ×

$2 \times 2 \times 1 \times 1 \times 1 = 2^2$





⇒ Sorted Array



Contribution of a_0 in the sum = $a_0 \times 2^0$

Contribution of a_1 in the sum = $a_1 \times 2^1$

Contribution of a_2 in the sum = $a_2 \times 2^2$

⋮

Contribution of a_{n-1} in the sum = $a_{n-1} \times 2^{n-1}$

$$\text{Sum} = \sum_{i=0}^{N-1} a[i] \times 2^i$$

Sort (a)

```
for( i=0; i< N; i++ ){
    sum += (a[i]x (1<<i));
}
```

TC : O(NlogN)

SC : Depend on the sorting Algo.
