Subset and Subsequence

Advance Classes : 26th April

New Advance Batch for fast learners : Next Wednesday

Wed: 7am

Fri : 7am

Sat : 8:30 Pm

Subsets & Subsequence

Advance Class: 26th April

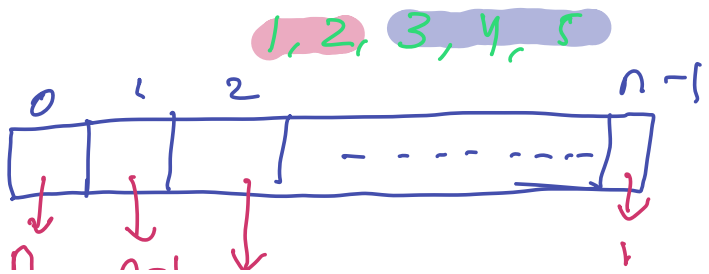
: 10 days

Problem solving classes

→ Bit Manipulation

→ Permutation / Combination

SubArray :



N-2

$\Rightarrow (n + (n-1) + (n-2) + \cdots$

$\Rightarrow \frac{n(n+1)}{2}$

$\Rightarrow$ **subsequence** sequence generated by deleting 0 or more elements from your array.

arr[8]: [3 -2 0 1 8 7 4 9]

× ✓ ✓ × ✓ × ✓ ×

{-2, 0, 8, 4} ✓ {1, 8, 9}

→ {8, -2, 4, 0} ×

{3}

{-2, 0, 1, 8}

{3, 9}

{-2, 0, 1, 8}

{3, 7, 8}

{

0 = 1, 2, 3, 4, 5

1, 2, 3, 4, 5

4

2, 3, 4

5, 4, 3

Sub array ⇒ subsequence ✓

→ subseq ⇒ subarray ✗

{1, 2, 3, 4, 5, 6}

→ {2, 4, 5}

⇒ {3, -2, 1} —sort→ {-2, 1, 3}

{ }	{ }
{3}	{3}
{-2}	{-2}
{1}	{1}
{3, -2}	{-2, 3}
{3, 1}	{1, 3}
{-2, 1}	{-2, 1}
{3, -2, 1}	{-2, 1, 3}

Subsets : like subseq, order does not matter

set : unique element

3 ← (3, -2, 1) ⟶ {-2, 1, 3}

All subset.

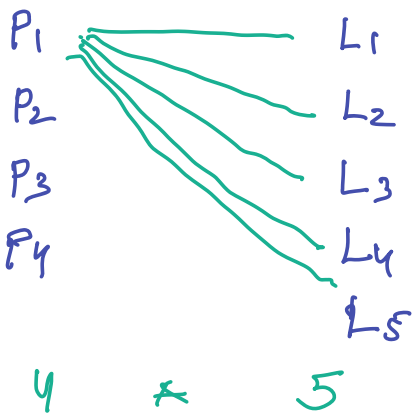
2^3 ⇒ 0

{ }	{ }
→ {3}	{3}

{-2}
{1}
→ {3, -2}
→ {3, 1}
{-2, 1}
→ {3, -2, 1}

{-2}
{1}
{-2, 3}
{1, 3}
{-2, 1}
{-2, 1, 3}

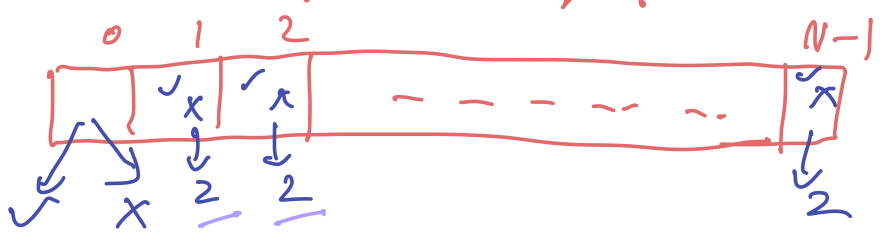
Valentine day



4 * 5

P1 : 5 choices
P2 : 5 —
P3 / P4 : 5 —
⇒ 20

Count # of subseq ?



2 * 2 * 2 * ---- * 2
n

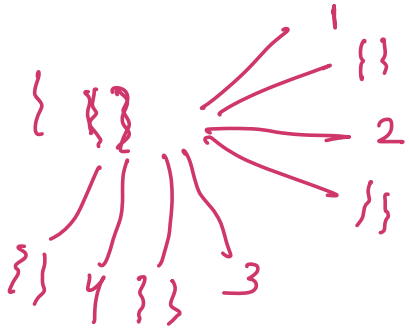
2^n

$2*2 = 2^2$

$2*2*2 = 2^3$

$2*2*2*2 = 2^4$

distinct elem ⇒ subsets ⇒ 2^n



{1, 2, 2}

subseq

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

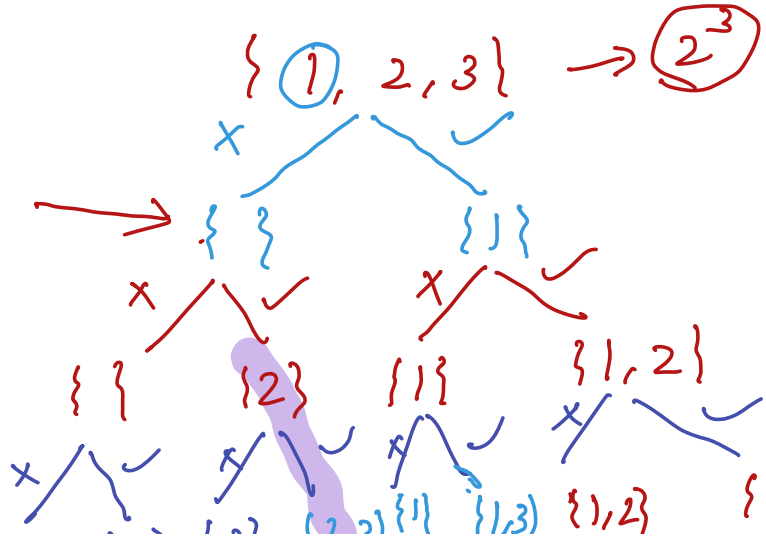
{1, 2}, {2, 2}, {1, 2}

{1, 2, 2}

{1, 2, 2}

subset

Break: 9:30



{1, 2, 3} → 2^3

{1} → 2^1 ⇒ 2

{1, 1}

{1, 2} → 2^2 ⇒ 4

{}, {1}, {2},

{1, 2, 3}

{} {3} {2} {2,3} ... {1,4}

Q ⇒ Given N distinct element → check if there exist a subset with sum = K

ex:

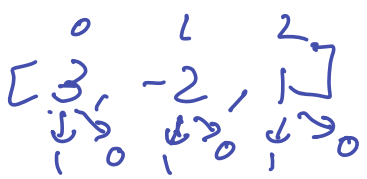
3, -1, 0, 6, 2, -3, 5

K=10

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

K = 20
No

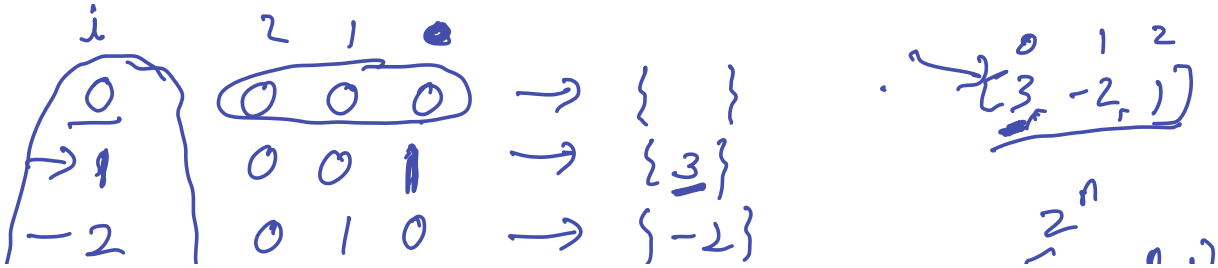
n-k

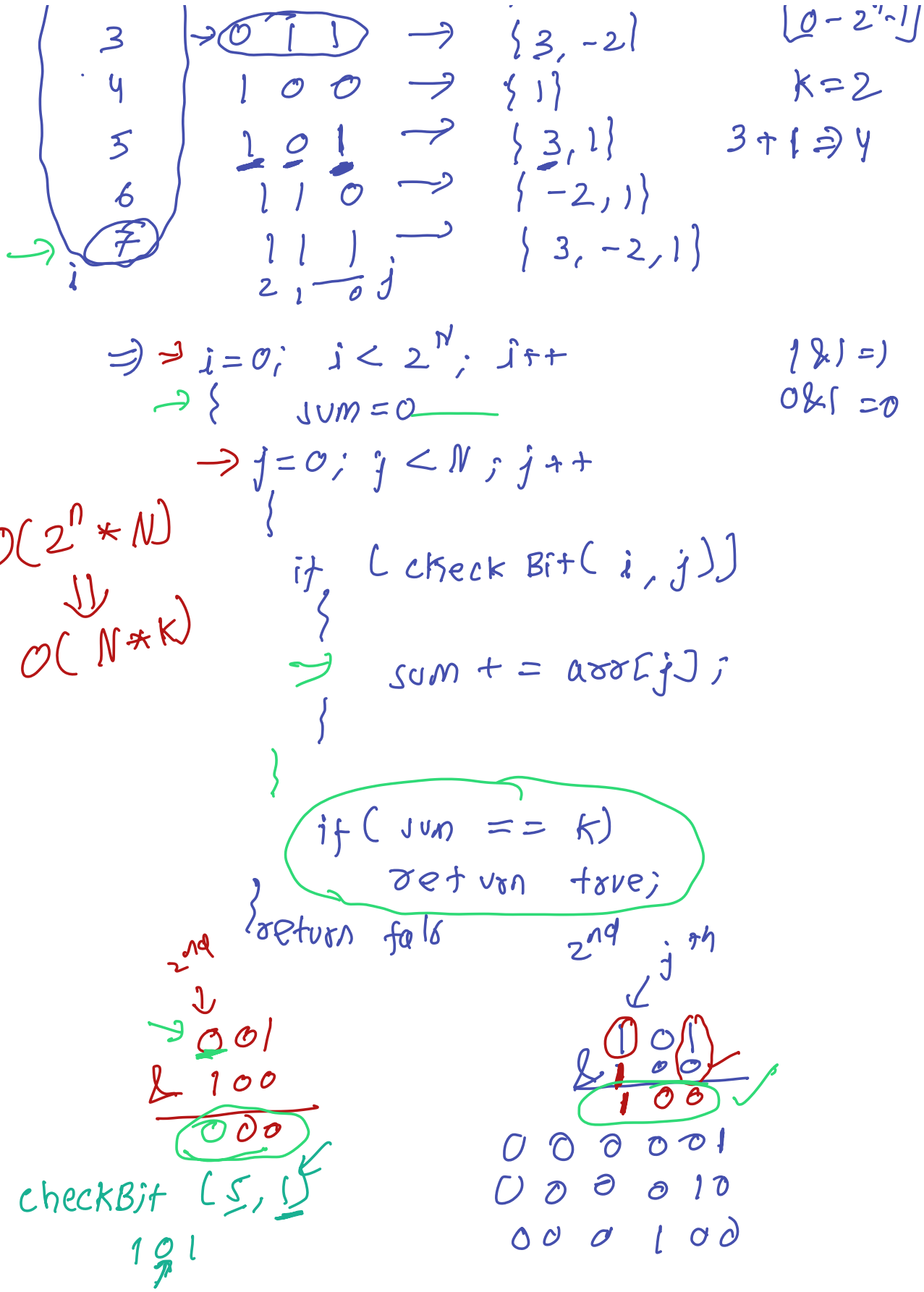


{3}

2 1 0
0 0 1
0 1 1 → {3, -2}
1 1 1 → {3, -2, 1}
1 0 1 → {3, 1}

$2^3 = 8$




3
4
5
6
7
i
0 1 1 → {3, -2}
1 0 0 → {1}
1 0 1 → {3, 1}
1 1 0 → {-2, 1}
1 1 1 → {3, -2, 1}
2 1 0 j
[0 - 2^n - 1]
K = 2
3 + 1 ⇒ 4
1 & 1 = 1
0 & 1 = 0
⇒ → i = 0; i < 2^N; i++
→ { sum = 0
→ j = 0; j < N; j++
{
if [checkBit(i, j)]
{
→ sum += arr[j];
}
}
if (sum == K)
return true;
return false
O(2^n * N)
O(N * K)
2nd
001
& 100
000
checkBit [5, 1]
101
2nd
j th
101
& 100
100
0 0 0 0 0 1
0 0 0 0 1 0
0 0 0 1 0 0

Q ⇒ Given n distinct element, sum of all subset.

{3, 1, 4}

{} → 0

{3} → 3

{1} → 1

{4} → 4

{3, 1} → 4

{3, 4} → 7

{1, 4} → 5

{3, 1, 4} ⇒ 8

32

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

Contribution Technique

⇒ 3*4 + 1*4 + 4*4

⇒ 32

{3, 2, 6, 8} → 2^4 ⇒ 16

1 2 2 2

→ 2^3 ⇒ 2^{4-1}

$3 * 2^3 + 2 * 2^3 + 6 * 2^3 + 8 * 2^3$

s = 0

O(N) i = 0; i < N; i++

s = s + arr[i] * (2^{N-1})