Subsequences and Subsets

May 9, 2022

AGENDA:

- Subsequences & Subsets
- Some interesting problems

$$\begin{array}{cccccc} & & 0 & 1 & 2 & 3 \\ N = & [& 3 & -1 & 2 & 6\,] \\ & & \multicolumn{4}{c}{\underbrace{\hspace{6em}}_{N \text{ elements}}} \end{array}$$

How many subarrays?

$$\left.\begin{array}{l} i \rightarrow 0 \text{ to } N \\ j \rightarrow i+1 \text{ to } N \end{array}\right\}$$

AP

No. of subarrays $= \dfrac{N(N+1)}{2}$

Subsequences

[3, 6]
[2, 3]

↳ need not be contiguous, but must be ordered.

Def: Generated by removing 0 or more elements from the array.

$A = [3, -1, 0, 2, 6]$

$A_i = [\,]$ ← // A subsequence.

$a = [3, -1, 0, 2, 6]$ ← // Yes.

$[3, 0, 6]$ ← Yes.

→ [2, -1] ← No ✗

** Order of elements in a subsequence is preserved.
Elements are placed in order of original indices.

* All subarrays are subsequences. ✓
All subsequences are subarrays.

[-3, $\underbrace{0, 1, 2}$, 9]

↓

[0, 1, 2] → subarray
↓
original indices are preserved.
& contigous.

subarray ✗✗ ← [-3, 2, 9] → ordered but not contigous

Subsets.

[-1, 0, 1, 3, 2]

[1, 0] → × Subseq ×
[1, 0] → ✓Subset ✓

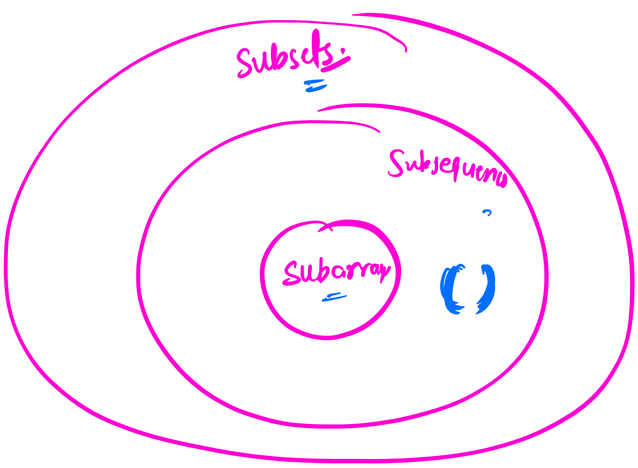
* Subarray — Contig. & ordered,
Subarrays can't be empty.

Subsequences — Ordered

Subset — ×.

1. All subarrays are subsets.
2. All subsets are subsequences.
3. All subsequences are subsets.
4. All subsets are subarrays.

Can subsets be empty? → Yes.



*Subarrays can't be empty.

a = [ ]
Empty subsequence.

Q.

[4, -1, 2]

[-1, 4] is a subsequence ✗

is a subset. ✓

Q. Will sorting the array affect the subsequences?

[4, -1, 2] $\xrightarrow{\text{sort}}$ [-1, 2, 4]

4	[4]	1	2	[-1]	-1
-1	[-1]	2	3	[2]	2
2	[2]	3	1	[4]	4
4	[4, -1]	4		[-1, 2]	2
4	[4, 2]			[2, 4]	4
2	[-1, 2]			[-1, 4]	4
4	[4, -1, 2]			[-1, 2, 4]	4

* Subsequences are affected by sorting.

sum

Q. Find the max of each subsequence.

Is it affected by sorting?

Q.1 Count no. of subsequences for a given array. of size N.

[1] → 2
N=1

[-1, 1] → 4
N=2

[ ]
-1
1
-1, 1

[0, 1, 2] → 8
N=3

[ ]
0
1
2
0 1
1 2
0 2
0 1 2

Total no. of subsequences = 2^N
non-empty = $2^N - 1$

N^2 $N!$

$[a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad \dots \quad a_n]$

Def: Generated by removing 0 or more element.

$[a_1/a_1' \quad a_2/a_2' \quad a_3/a_3' \quad \dots \quad a_n/a_n']$

$2 * 2 * 2 * \dots\dots 2$

→ 2^N possible combinations

Total no. of subsequences $= 2^N$

$2^N - 1$ (not empty)

Q. $[-2, 0, 3]$

Are $[0, 3]$ & $[3, 0]$ different subsets?

** $\boxed{\text{Same}}$ **

How many subsets are present for array of size N?

No. of subsets $= 2^N$

* If there are repeated elements?

[2, 2, 1, 0, 0]

[2, 2] ?

[2, 2] → Subarray . . ✓

[2, 2] → Subsequence

[2, 2] → Subset . ✓

t = set() × Different

[2, 2]
[2, 0, 0]
[0, 0]

Different
* [2, 1] — ×
* [2, 1] — ×

* [2, 2, 0] ×

Bool

Q. Given an array of N elements, check if there is a subset with given sum X.

**

Subsequence

A = [3, -1, 0, 6, 2, -3, 5]

X = 10

6 + 5 + (-1)

5 + 2 + 3.

3 -1 6 2.

Brute Force.

* $\underbrace{\text{Find all subsets}}_{O(2^N)}$, $\underbrace{\text{get their sum}}_{O(N)}$

$\underbrace{O(N * 2^N)}$

Brute force.

Part 1

* Generate all subsets.

A = [-2, 6, 4]

How many subsets = $2^3 = 8$

0/1 0/1 0/1

Bit m

picked			
x	0 0 0	→	0
[4]	0 0 1	→	1
[6]	0 1 0	→	2
[6,4]	0 1 1	→	3
[-2]	1 0 0	→	4
[-2,4]	1 0 1	→	5
[-2,6]	1 1 0	→	6
[-2,6,4]	1 1 1	→	7

8

** No. of elements in the array (N) = No. of bits in my no. (M)

2^N $\qquad$ 2^M

✓ x ✓
[0, 0, 1]
↓
[0, 1]

101 ← 5

A
N = no. of elements in array

```
for num in range (0, 2**N):
        # map num to pick/drop elements from
            the array.
        subset = [],  subsetSum = 0
        for idx in range (0, N):
                if checkBitISet (num, idx):
                        subset.append( A[idx])
```

AND 2^i

subsetSum += A[idx]

if subsetSum == x :
    return True

return False.

end 1st 0th

101

[-2, 0, 1]

N = 3

0 to $2^3 - 1$

0 → 000 → subset [ ]

1 → 001 → [1] 1

2

3 → 011 → [1, 0] 1

4 → 100

5

6 → 110 → [0, -2] -2

7

$$
\begin{array}{rl}
\text{AND} & 11001111 \\
 & 00010000 \\
\hline
 & 00010000
\end{array}
$$

← 2^i mask

Break till 10:17

Can we optimise ?

$O(N*2^N) \longrightarrow O(2^N) \longrightarrow O(2^{N/2})$

(Backtracking)

(Meet in the middle)

$O(N^2) \leftarrow O(N*X)$

Dynamic program

similar (0-1 Knapsack problem)

Q. Given N array elements, find the sum of all subset sums.

[1, 2] →

[1] → 1

[2] → 2

[1, 2] → 3

Ans = 6

[-1, 1] → 0

Brute force

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

Optimise?

TC: $O(N)$

* Sum of all subarray sum

$O(N)$

→ $1a_1 + 2a_2 + 3a_3 + 4a_4 + 5a_5$

N = [1, 2, 3]

In how many subsets 1 is present?

Ans = 4

[1, 2]
[1, 3]
[1, 2, 3]
[1]

" subsets 3 is present?

Ans = 4

N [1, 2]

In how many subsets 1 is present → 2

A = [5 3 -1 0 2]

3 -1 0 2 → $\boxed{2^{n-1}}$

5 →

**

[5 3 -1 0 2]

1 2 2 2 2

$2^4 = 16$ ✓

In a given array, (of size N)

A[i] will be present in how many subsets?

$= 2^{N-1}$

A [1, 2, 3]

[1] = 1
[2] = 2
[3] = 3
[1, 2] = 1+2
[2, 3] = 2+3
[1, 3] = 1+3
[1, 2, 3] = 1+2+3

$= (1+2+3)^{*4}$

* Sum of all subset sums.

$A[i] * \underbrace{2^{N-1}}$ $\forall i$

```
{ p = pow(2,N-1)
  s = sum(A)
  return p*s
```

→ TC: $O(N)$

Q. Given N array elements, find the sum of max of every subsequence.
↳ subset.

[3, 1, -4]

3	→	3
1	→	1
-4	→	-4
3, 1	→	3
-4, 1	→	1
3, -4	→	3
3, 1, -4	→	3
		10

$$\begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ -4, & 1, & 3, & 6, & 8 \\ \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\ 2 & 2 & 2 & 1 & 1 \end{array}$$

* In how many subsets -4 is the max? 0

① subset 2^0

* " 1 is the max? 1

-4, 1 ② 2^1

1

* " 3 is the max ④ 2^2

* " 6 is the max ⑧ 2^3

Contribution of $\underbrace{A[i]}_{\text{(sorted array)}} = 2^i * A[i]$

Total sum of all max $= 2^i * A[i] \;\forall i$ in a sorted array.

TC: $\rightarrow$ O(N log N)

(Sort the array!)

** Contribution technique **

Sum of all sums.

Sum of maxe

Sum of all subarray sum

Q. Find the sum of minimum of each subsequence.

* Sort the array in descending.
* $A[i] * 2^i$ 2^{N-i-1} if ascending.

Q. Find sum of (max - min) of every subsequence.

Total max - Total min

$(max_1 - min_1) + (max_2 - min_2) + (max_3 - min_3)$

$\vdots$

$= \underbrace{(max_1 + max_2 + \dots max_n)}_{Ans1} - \underbrace{(min_1 + min_2 + \dots min_n)}_{Ans\ 2}$

Q. Calculate $\dfrac{\text{sum of all subset sums}}{2^N}$

$$= \frac{sum(A) * 2^{N-1}}{2^N}$$

$$= \frac{sum(A)}{2}$$

** **Gray Code / Kth symbol**

$\boxed{OOP}$

Decimal	Binary	Gray code	
0	000	000	0
1	001	001	1
2	010	011	3
3	011	010	2
4	100	110	6
5	101	111	7
6	110	101	5
7	111	100	4
8	1000		
9	1001		
10			
11			

from 1 to 10

$\boxed{0, 1, 3, 2, 6, 7, 5, 4}$

* $\boxed{00 \to 01 \to 11 \to 10}$ ✓
* 11 → 10 → 00 → 01 ✓

10 → 11 → 01 → 00

2 bits

0	000	→ 0	0 ←	
0	001	→ 1	1 ← 1+0 = 1	
0	011	→ 3	1+2 = 3	
0	010	→ 2	0+2 = 2	
0	110	→	2+4 = 6	
0	111	→	3+4 = 7	
0	101	→	1+4 = 5	
0	100	→	0+4 = 4	
→	1100		4+8 = 12	
	1101		5+8 = 13	
	1111		7+8 = 15	

1110	$6+8=14$
1010	$2+8=10$
1011	$3+8=11$
1001	$1+8=9$
10000	$0+8=8$

Q.

1 0
2 0 1
3 0 1 1 0
4 0 1 1 0 1 0 0 1
5 0 1 1 0 1 0 0 1 1 0 0 1 0 1 1 0

**

$2^{N-1}+1 \rightarrow 1$
$2^{N-1}+2 \rightarrow 2$

Bth bit in Ath row. = 2^N

if $B < \frac{2^N}{2} \left(2^{N-1}\right)$
ans = $(A-1)^{th}$ row's Bth Bit

if $B > 2^{N-1}$
ans = inverse of $(A-1)^{th}$ row
$B - 2^{N-1}$ Bit in $A-1^{th}$ row.

1 0
2 0 1
3 0 1 1 0
4 0 1 1 0 1 0 0 1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
5	0	1	1	0	1	0	0	1	1	0	0	1	0	1	1	0

$12 - 8 = 4$

Bth bit in 5th row.

||

Bth bit in 4th row

Bth bit in (A-1)th row.

Bth bit is in 2nd half.

ans = inverse of corresponding bit in 1st half $- \frac{2^N}{2}$