

l r
 1 3 \leftarrow n length
 2 5
 1 2
 5 6

1	2	3	4	5	6	7
2	1	-1	-1	1	-1	-1

$\leftarrow a$

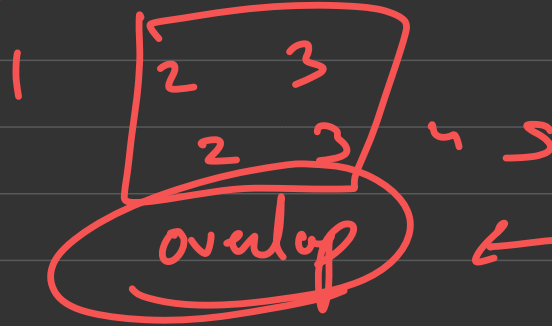
Difference array / vector

To process 1 query \rightarrow $O(n)$ / query

$a[l] += 1$
 $a[r+1] -= 1$

Prefix Sum

anyhow I can store the start & end of any query



\leftarrow repetitive manual

Prefix
Sum

1	2	3	4	5	6	7
2	1	-1	-1	1	-1	-1

← a

2	3	2	1	2	1	0
---	---	---	---	---	---	---

✓✓

1	2	3	1	0	0	0	0	0	0	1	0
0	1	2	3	4	5	6	7	8	9	10	

✗ freq

1
2
4
2

Suffix
Sum

7	6	4	1	0	6	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---

allent: i coins

i coins + allent(i+1) coins

$$f(i) = g(i) + \underline{f(i+1)}$$

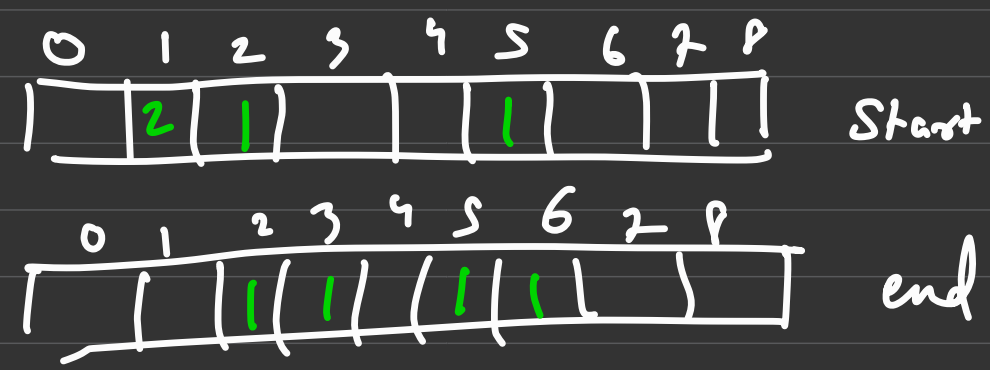
\swarrow
at least
i coins

\swarrow
exact i coins

\downarrow
at least (i+1) coins

approach 2

2 7
 1 3
 2 5
 1 2
 5 6



at least i coins

1st box $\rightarrow \text{start}[1] \rightarrow 2$

2nd box $\rightarrow \text{start}[1] + \text{start}[2] - \text{end}[1]$

3rd box $\rightarrow s[1] + s[2] + s[3] - e[2] - e[1]$

4th box $\rightarrow s[1] + s[2] + s[3] + s[4] - e[5] - e[2] - e[1]$

1st box $\rightarrow \text{start}[1] \rightarrow 2$

2nd box $\rightarrow \text{start}[1] + \text{start}[2] - e$

3rd box $\rightarrow s[1] + s[2] + s[3] - e[2] -$

4th box $\rightarrow s[1] + s[2] + s[3] + s[4] - e[3] - e[2] - e[1]$

$t(i) \rightarrow$ coins in i^{th} box $+ [i] = s[i] + t[i-1] - e[i]$

$$+ [1] = s[1]$$

$$+ [2] = s[1] + s[2] - e[1] \Rightarrow s[2] + + [1] - e[1]$$

$$+ [3] = s[1] + s[2] + s[3] - e[2] - e[1] \Rightarrow s[3] + + [2] - e[2]$$

$$+ [4] = s[1] + s[2] + s[3] + s[4] - e[3] - e[2] - e[1] =$$
$$s[4] + t[3] - e[3]$$

LIS ✓ (largest inc subsequence)

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

Brute force → Calc all the subsequences, then filter

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

$f(i) = 1, 2, 2, 3, 2, 3, \dots$

$$f(i) = 1 + \max(f(j)) \quad \forall j \in [0, i-1] \text{ and } \underline{\underline{a[j] < a[i]}}$$

\downarrow

returns LIS
ending at i

Time $O(n^2)$
Space $\rightarrow \underline{\underline{O(n)}}$

result = $\max(f(i))$ $\forall i \in [0, n-1]$

1, 2, 1, 5, 4

1, 2, 1, 3, 3

ans - 3

[1, 2, 5]

Lis = 1 2 2 3 2 3 3 4 2 4 3 5 3 5 4 6

$[0]$

Q²

[2, 4, 6, 8, 10]

Arithmetic \rightarrow atleast 3 elem
subseq

\rightarrow forms AP

AP diff

$f(i)$ = returns the count of AP subsequn
of atleast length 2.

(2, 4, 6, 8, 10)
 x (2-1) 4-1 6-1 8-1
 2-2 4-1 6-2
 2-3 6-1
 8-1

a way of heapup
 →

$$f(i, d) = 1 + f(k, d)$$

$$\forall k \in \underline{\underline{[0, i-1]}}$$

⇒ longest Biotonic subarray (inc → dec)

[1, 2, 3, 3, 2]

[1, 11, 2, 10, 4, 5, 2, 1] ←

forward _{lis} → [1, 2, 2, 3, 3, 4, 2, 1]

backward _{lis} → [1, 5, 2, 4, 3, 3, 2, 1]

Basic C++

(lay)

Reading

plus
Spaw

Recursion

Backtracking

sorts & search →

Binary Search

Base DSA

LL, ST, queue, trees, HM.

additional C++