2. Arrays and Bit Manipulation

09 June 2024 13:52

Arrays → collection of similar data type items stored at continuous memory location

int a[5];

1000	
1004	
1008	
1012	
1016	

⇒ Q/. Given array a. Find sum of subarray [l, r].

sum = 0
for (i = l ; i <= r ; i++)
sum = sum + a[i]

TC = O(N)

Q/. There are Q queries
q_i → (l_i, r_i) →

a = [3, 2, 1, 5, 7, 9, 10, 11]   n = 8

q = 4 → [0, 3] → 11
[4, 6] → 26
[2, 7] → 43
[1, 5] → 24

Brute force :-
for each query i
loop [l_i, r_i]

make efficient ← TC = O(Q×N)

a = [3, 2, 1, 5, 7, 9, 10, 11]
ps = [3, 5, 6, 11, 18, 27, 37, 48]

ps[i] = [0, i]
ps[r_i] − ps[l_i − 1]
s[ ] + a[i] ← + a[r_i]

$ps = [3, 5, 6, 11, 18, 27, 37, 48]$

$[3, 6]$

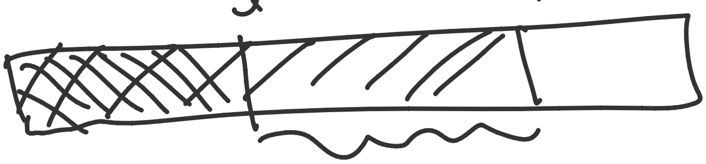
$ps[6] = 37$
$ps[2] = 6$ $\rightarrow 31$

$ps[r_i] - ps[l_i - 1]$

$a[0] + a[1] + \_\_\_ + a[r_i]$

$-(a[0] + a[1] + \_\_\_ + a[l_i - 1])$

$= a[l_i] + a[l_i + 1] \_\_\_ + a[r_i]$



$TC = O(N + Q)$

$N = 10^5, \; Q = 10^5$

10^{10} 10^5

$ps[i] = ps[i-1] + a[i]$

$ps[i] = a[0] + a[1] + \_\_\_ + a[i]$

$ps[i-1] = a[0] + a[1] + \_\_\_ + a[i-1]$

$ps[0] = a[0]$

for $i = 1$ to $N-1$:
  $ps[i] = ps[i-1] + a[i]$ $\rightarrow N$

$N+Q$

for $i = 0$ to $Q-1$:
  $ans = ps[r_i] - ps[l_i - 1]$ $\rightarrow Q$

$q = 3$

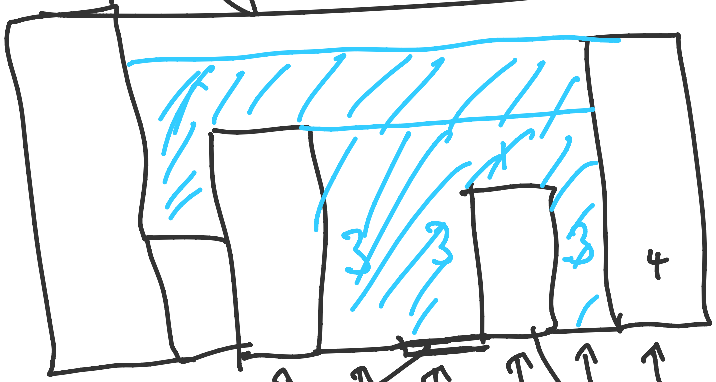
$queries[] = \{\underbrace{1, 3}_{i=0}, \underbrace{2, 7}_{i}, \underbrace{3, 5}\} \rightarrow 6$

$q = 5$ 10

29

Trapping rain water

[3, 0, 0, 2, 0, 4]



10

min(3,4) - 2 = 1

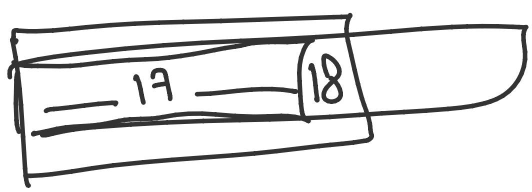
min(3,4)

min(5,4) = 4

```
{ for i = 0 to n-1:
sum = sum + min(lm[i], rm[i]) - arr[i]
}
```

lm[i] → pmax[i], rm[i] → suffix max[i]

pm[i] = max(pm[i-1], a[i]);



Q/. Maximum sum K size subarray.

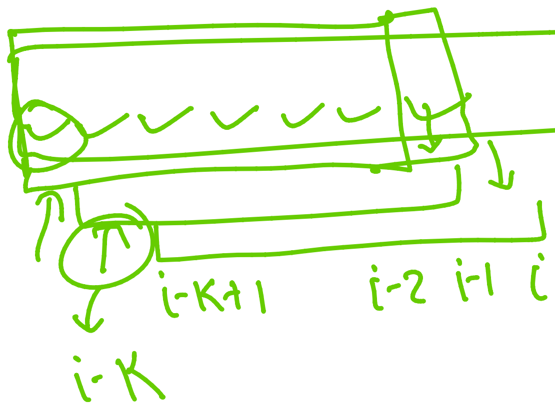
a = [3, 1, 9, 12, 6, 18, 2, 1, 20, 21]

K = 4

n-K+1

25 28 40 38 27 41 44

40 38 27 41 44

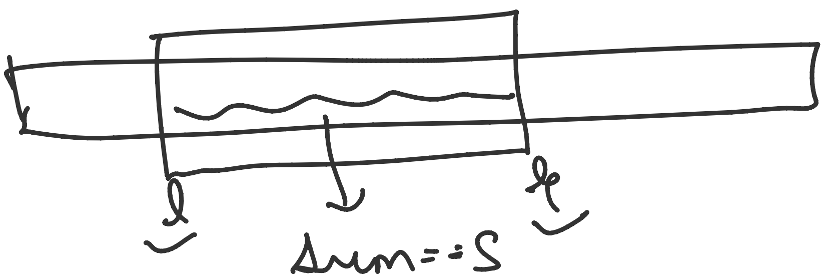


Sliding window

Q/ Subarray with given sum :-

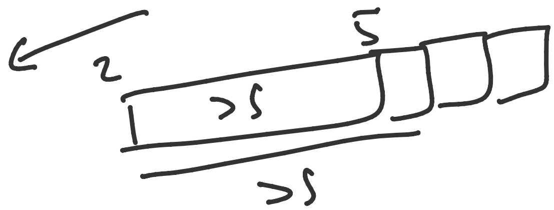
array a → n

S → target

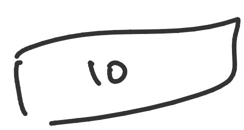


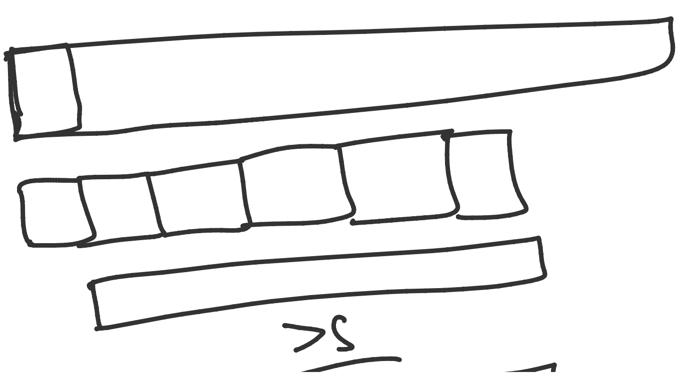
Efficient sol^n :-

Brute force → $\frac{n(n+1)}{2}$

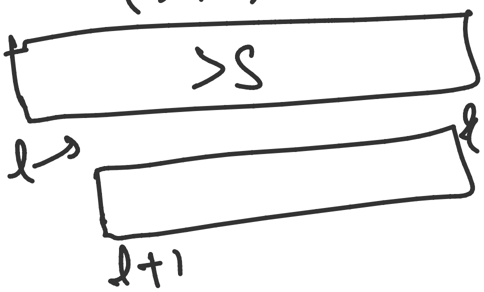


tar = 18



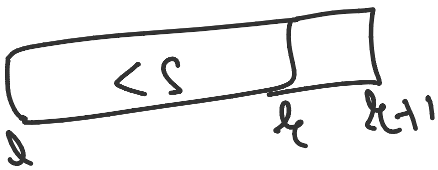


$(l, r) > S$



$(l, r+1)$
$(l, r+2)$
$(l, r+3)$
$\vdots$
$(l, n-1)$

$> S$

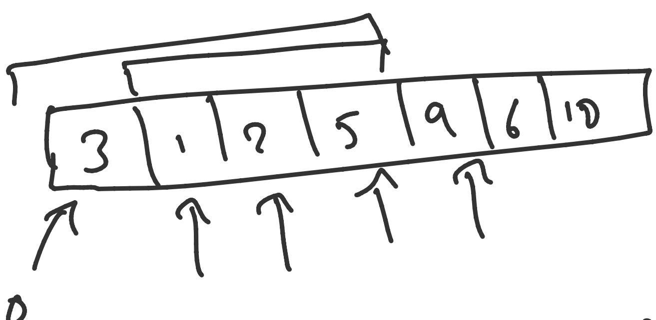


$(l+1, r)$
$(l+1, r-1)$
$(l+1, r-2)$
$\vdots$
$(l+1, l+2)$

$(l+2, r)$
$(l+2, r-1)$
$\vdots$
$(l+2, l+2)$

$< S$

→ 2 pointers (l, r)



$S = 10$

sum = 3, 4, 6, 11, 8, 17

for i = 0 to N-1:
 for j = 0 to N-1:

$i = 0$ ——— $N-1$

N N N N

for j = 0 to n-1:

$l = 1, r = 0$

0 ↓ ↓1

$S = 0$, 1   {1, 2, 3, 4}

sum = 1, 0

{1, 2, 0, 3}

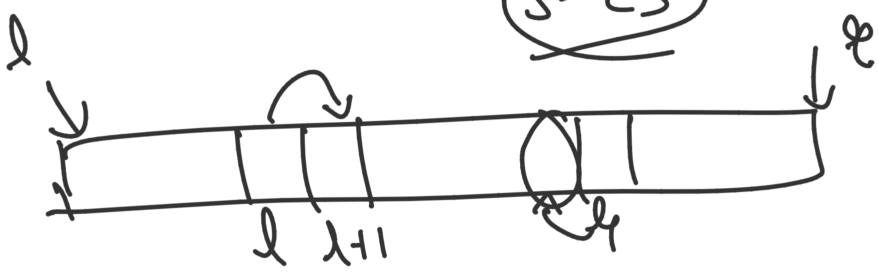
2, 2

sum = 0

Q Given a sorted array. Find whether there exists a pair of elements (i, j), $i \neq j$ and $a[i] + a[j] = S$.

{2, 5, 9, 12, 18}

$S = 23$



$a[l] + a[r] < S$

$a[l] + a[r] > S$

$l = 0, r = n-1$

while ($l < r$)

{ if ($a[l] + a[r] > S$)

TC = O(N)

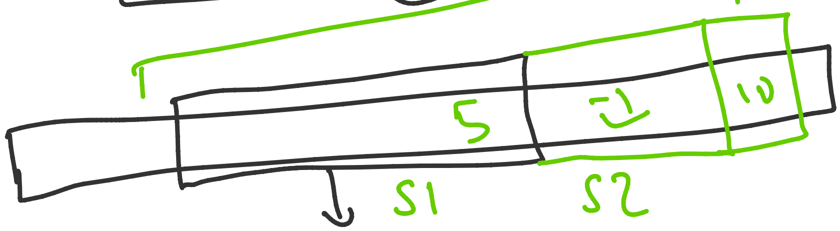
2 pointer

while (...)
{ if (a[l] + a[r] > s)
r--
else if (a[l] + a[r] < s)
l++
else return <l, r>
}

2 pointers

Q1.

(1, 2, -1, 10) → 12



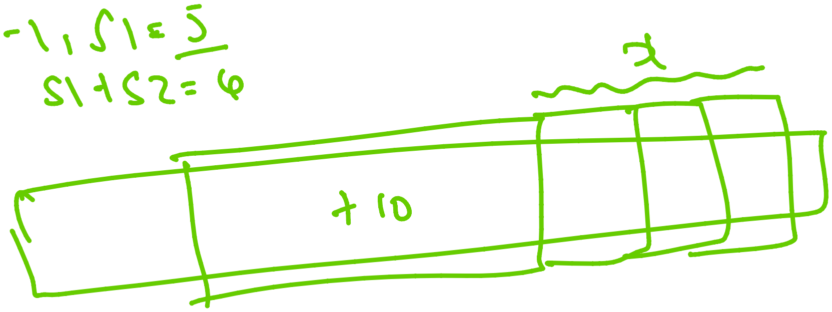
sum > 0
< 0
= 0

S1 + S2

S2 = 10, S1 = 5
S1 + S2 = 15

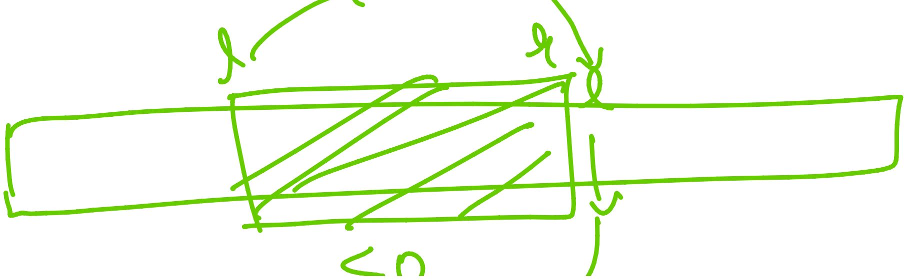
S1 = -5, S2 = 100 → 95
S2 = -1 → -6

S2 = -1, S1 = 5
S1 + S2 = 4

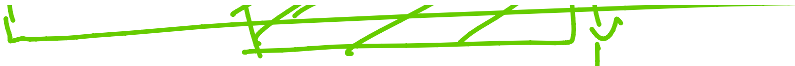


> 0

ans = max(ans, S1)



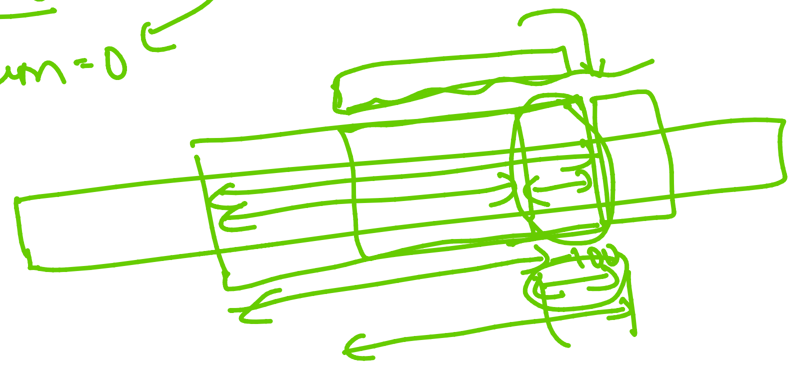
< 0



≤ 0

sum = 0

Kadane's Algo



Initialization :- Always initialize ans with one of the possible answers.

Q/ Given multiple ranges, you need to return maximum occurring element.

→ [1, 7] [4, 9] [2, 6] [10, 13]

10, 11, 12, 13 → 1

1 → 1
2 → 2
3 → 2
4 → 3

5 → 3
6 → 6
7 → 2
8 → 1
9 → 1

n ranges
(l_i, r_i) c(i)

→ c(14)

0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	$\cancel{0}$ 1	$\cancel{0}$ $\cancel{1}$ 2	$\cancel{0}$ $\cancel{1}$ 2	$\cancel{0}$ $\cancel{1}$ $\cancel{2}$ 3	$\cancel{0}$ $\cancel{1}$ $\cancel{2}$ 3	$\cancel{0}$ $\cancel{1}$ $\cancel{2}$ 3	$\cancel{0}$ $\cancel{1}$ 2	$\cancel{0}$ 1	$\cancel{0}$ 1	$\cancel{0}$ 1	$\cancel{0}$ 1	$\cancel{0}$ 1	$\cancel{0}$ 1

$r_i - l_i \leq 10^5$

$n \leq 10^5$

$TC = O(10^{10})$

c(l)++

$n \le 10^5$

$[1, 7]$ $[4, 9]$ $[2, 6]$ $[10, 13]$ $c[l]{+}{+}$
$c[r+1]--$

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	$\cancel{0}$ +1	$\cancel{0}$ +1	0	$\cancel{0}$ +1	0	0	$\cancel{0}$ -1	$\cancel{0}$ -1	0	$\cancel{0}$ $\cancel{-1}$ 0	0	0	$\cancel{0}$ -1	
0	1	2	2	3	3	3	2	1	1	1	1	1	1	0

$O(10^5)$

$[2, 6]$

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

Bit Manipulation :

Here, we use techniques involving bitwise operators.

$\underbrace{\&,\ |,\ >>,\ <<,\ \wedge}$

$a = 10, 12, 13$. --

13 → base 10

$1 \times 10^1 + 3 \times 10^0$

213 → 2×10^2

Binary representation

$10 \longleftrightarrow 1010$

$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$

$1\times2 + 0\times$ ...

$10 \longleftrightarrow 1010$

$2^3 \quad 2^2 \quad 2^1 \quad 2^0$

$+ 0\times2^0$

$= 8+0+2+0$

$= 10$

10 ⟶

2	10	0
2	5	1
2	2	0
2	1	1
	0	

1010

① AND (&) :-

10 & 9 = 8

0	0	0
0	1	0
1	0	0
1	1	1

$$\begin{array}{r} 1010 \\ 1001 \\ \hline 1000 \end{array}$$

10 & 9 & 8

$$\begin{array}{r} 1010 \\ 1001 \\ 1000 \\ \hline 1000 \end{array}$$

② OR (|) :-

a = 10, b = 9

a|b = 11

0	0	0
0	1	1
1	0	1
1	1	1

$$\begin{array}{r} 1010 \\ 1001 \\ \hline 1011 \end{array}$$

3. ^ (XOR)

$a = 10, b = 9$

$a \wedge b = \underline{3}$

```
 1 0 1 0
 1 0 0 1
 -------
 0 0 1 1
```

0	0	0
0	1	1
1	0	1
1	1	0

$a = 10, b = 9, c = 8, d = 11, e = 7$

```
 1 0 1 0
 1 0 0 1
 1 0 0 0
 1 0 1 1
 0 1 1 1
 -------
 0 1 1 1
```

$a \wedge b \wedge c \wedge d \wedge e = \underline{7}$

4. << (left shift)

$a << b$

0 1010 → 10

10100 → 20

$c = a << b$ →

$\boxed{c = a \times 2^b}$

$a = 10, \quad b = 3$

0001010 → 10

0010100 → 20

0101000 → 40

1010000 → 80

80

5. >> (right shift)

~~1010>>1~~

10>>1

1010 → 10

→0101 → 5

$c = a >> b$

a 1

5>>1

→ 101 → 5

$c = a >> b$

$c = \left\lfloor \frac{a}{2^b} \right\rfloor$

$5 >> 1 \rightarrow 101 \rightarrow 5$

$\rightarrow 010 \rightarrow 2$

10 → MSB → 1 0 1 0 1 1 1 0 → LSB

2^P ... 2^0

- - - - - - - 1
even $2^0(1)$

odd → LSB → 1
even → LSB → 0

13 → 1 1 0 1
1 1 0
1 1
1

1 0 1 1
$\boxed{1101}$

```
while(n>0)
{ if(n%2 ==1)
    add → '1'
  else
    add → '0'  → n = n/2
}
reverse
```

Q/. K^{th} bit

K=4

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

&

0 0 0 0 1 0 0 0 0

0 0 0

2^K

K=4

return n & (2K) ; n = 1 1 0 1 0 1 0 1 0

2K = 0 0 0 0 1 0 0 0 0
&

$2^k = \underbrace{0000}1\underbrace{0000}$
$\&$
$\quad 0 \quad 0 \quad 0$

Power of 2 :-

$2^0 \to 1 \to 1$
$2^1 \to 2 \to 10$
$2^2 \to 4 \to 100$
$2^3 \to 8 \to 1000$
$2^4 \to 16 \to 1000$

```
c = 0
while (n>0)
{  if(n%2==1)
      c++;
   n = n/2;
}
if (c==1) true
   else false
```

$\Rightarrow n = n \& (n-1)$

lowest
set bit → unset

1010
1001

$n = 110\,1\,0 \to 26$
$n-1 = 11001$
11000

$n = 1010$
$n-1 = 1001$
1000

$(n \& (n-1))$

$(n\&(n-1)) == 0$

Q/. Count no. of set bits in n.

```
c = 0
while(n>0)
{
    c++;
    n = n&(n-1);
}
```

```
c = 0
while (n>0)
{  if(n%2==1)
      c++;
   n = n/2;
```

3

$a \wedge a = 0$

$a \wedge 0 = a$

$$\begin{array}{r} 1110010 \\ 1110010 \\ \hline 0000000 \end{array} \qquad \begin{array}{r} 11010 \\ 00000 \\ \hline 11010 \end{array}$$

{2, 3, 3, 4, 4, 4, 4, 5, 5, 2, 2}

0

2

Q/. Two odd occurring elements :-

Given an array of n elements where every element occurs even no. of times except two numbers. Print those two numbers.