

Remove and add List

Next Greater

Alex and treasures

Remove and add List

Given two linked lists: A and B of size N and M respectively, and two integers C and D. Remove linked list A's nodes from the Cth indexed node to Dth indexed node (0 based indexing) and put linked list B in its place.

C = 2

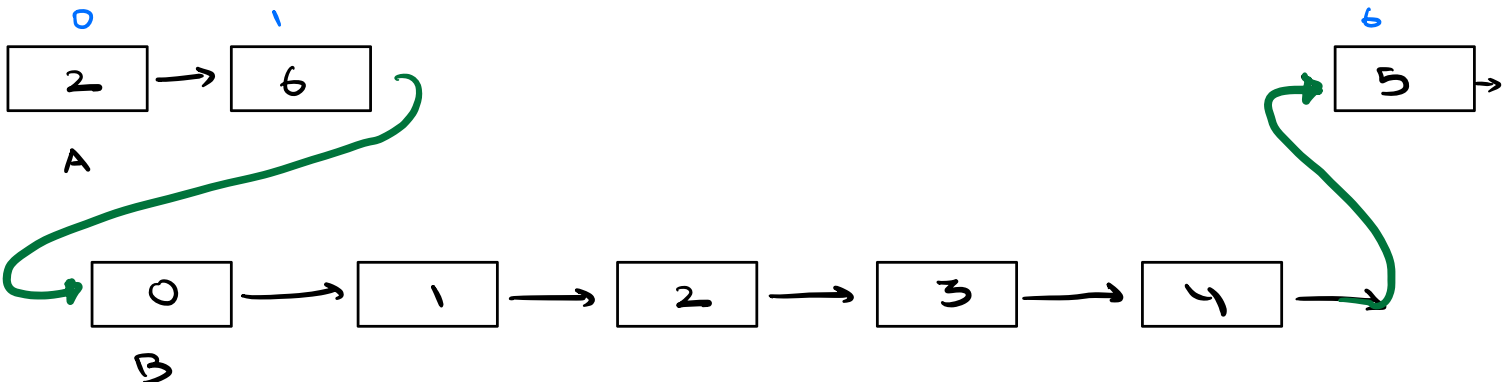
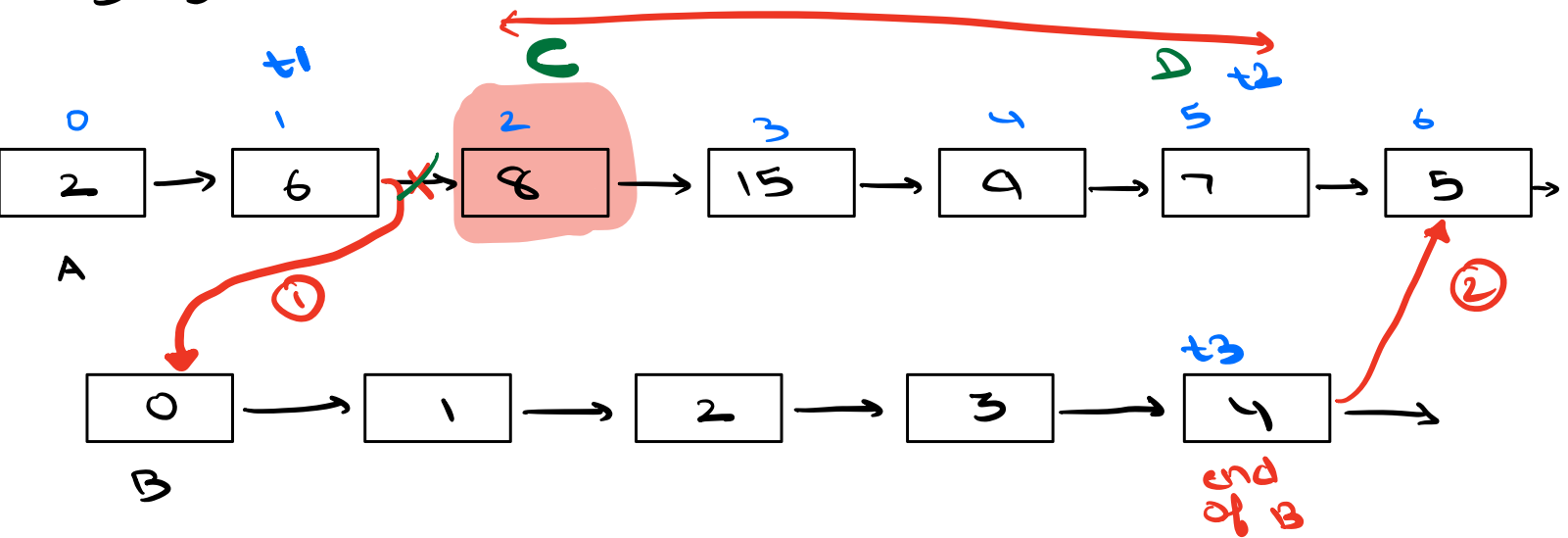
D = 5

A
N = 7

B
M = 5

$0 \leq N, M \leq 10^5$

$1 \leq C, D \leq N-1$



① Manipulate link of C-1

↓
make C-1 jumps → t1

t1.next = B

② Reach Dth node

↓
make D jumps → t2

end of B . next → D+1

③

t3 = B

Reach last node of B = t3

t3.next = t2.next
↓
D node

Node t1 = A , t2 = A

for (j = 1 ; j ≤ C-1 ; j++) {
| t1 = t1.next
| }

t1 → Node C-1

for (j = 1 ; j ≤ D ; j++) {
| t2 = t2.next
| }

t2 → Node D

①

$t1.next = B$

Node $t3 = B$

while ($t3.next \neq \text{NULL}$)

$t3 = t3.next$

3

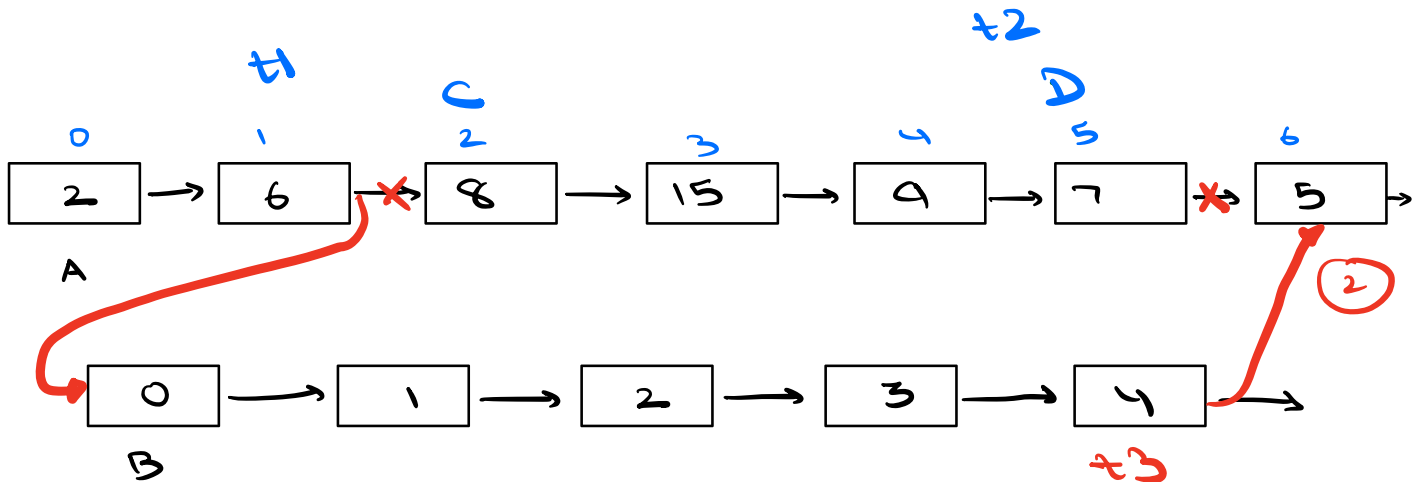
// $t3 \rightarrow$ last node of B

②

$t3.next = t2.next$

③

$t2.next = \text{NULL}$



TC : $O(N + M)$

SC : $O(1)$

Next Greater

Given an array A, find next greater element for every element of array.

Next greater element of $A[i]$ is first greater element on right side of $A[i]$.

If no greater element, consider it -1.

$A = [4, 5, 1, 2, 10]$

ans = $[5, 10, 2, 10, -1]$

$A = [3, 2, 1]$

$[-1, -1, -1]$

BF: For every ele, iterate on right side and find greater

TC: $O(N^2)$

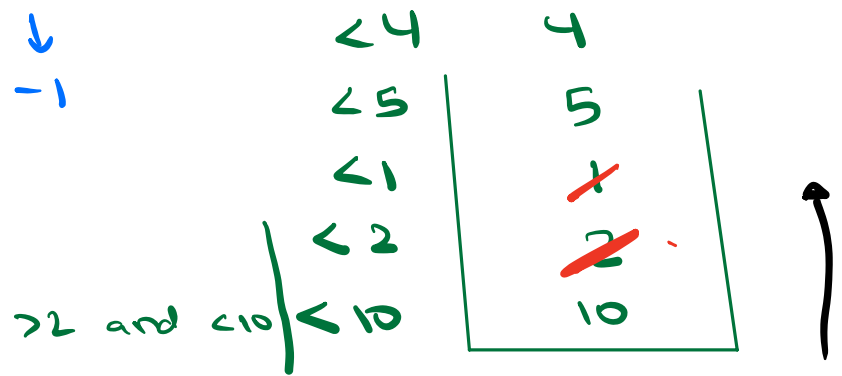
SC: $O(1)$

```
for (i = 0 ; i < N ; i++) {  
    // A[i]  
    for (j = i + 1 ; j < N ; j++) {  
        // ...  
    }  
}
```

Optimised:

$A = [4, 5, 1, 2, 10]$

↓ ↓ ↓ ↓ ↓
5 10 2 10, -1



Data in decreasing order
Delete smaller or equal elements

int ans[N] stack <int> st

for (i = N - 1 ; i ≥ 0 ; i--) {

while (!st.empty() &&
A[i] ≥ st.top()) {
 st.pop();

if (st.empty()) {
 ans[i] = -1;

else
 ans[i] = st.top();

st.push(A[i]);

}
return ans

TC: $O(N)$

SC: $O(N)$

Alex and Treasures

Given an array A of points where $A[i] = [x_i, y_i]$ represents a point on x - y plane and an integer B , return B farthest points to the origin $(0,0)$.

$$\text{Distance b/w 2 points} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

$$\begin{aligned} 1 &\leq A.size() \leq 10^5 \\ -10^4 &\leq x_i, y_i \leq 10^4 \\ 1 &\leq B \leq A.size() \end{aligned}$$

$$A = [[1, 2], [1, 4], [-1, 0]]$$
$$B = 2$$



$$A = [[1, 5], [2, 1], [-1, -1], [3, 1]]$$
$$B = 3$$

$$x_1, y_1 \rightarrow 0, 0$$

$$d = \sqrt{(x_1 - 0)^2 + (y_1 - 0)^2}$$

$$d = \sqrt{x_1^2 + y_1^2}$$

$$A = [[1, 2], [1, 4], [-1, 0]]$$
$$B = 2$$

$$\begin{aligned} 1,2 &\rightarrow \sqrt{1^2 + 2^2} = \sqrt{5} \\ 1,4 &\rightarrow \sqrt{1^2 + 4^2} = \sqrt{17} \\ -1,0 &\rightarrow \sqrt{(-1)^2 + 0^2} = \sqrt{1} \end{aligned}$$

① Sort as per distance from origin

$\sqrt{17}$	$\sqrt{5}$	$\sqrt{1}$
1,4	1,2	-1,0

O/P $\rightarrow (1,4) \quad (1,2)$

② Arranging pts in ans \rightarrow increasing dist X

Expected ans $\rightarrow (1,2) \quad (1,4)$

$\downarrow \quad \downarrow$
 $\sqrt{5} < \sqrt{17}$

$$\begin{aligned} \sqrt{1^2 + 5^2} &= \sqrt{26} & \sqrt{-1^2 + -1^2} &= \sqrt{2} \\ \uparrow & & \uparrow & \\ A &= [[1,5], [2,1], [-1,-1], [3,1]] \\ B &= 3 \end{aligned}$$

$$\begin{aligned} \downarrow & & \downarrow & \\ \sqrt{2^2 + 1^2} &= \sqrt{5} & \sqrt{3^2 + 1^2} &= \sqrt{10} \end{aligned}$$

$\sqrt{26}$	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$
1,5	3,1	2,1	-1,-1

$\sqrt{x^2 + y^2}$

↓
Arrange in inc order of dist

O/P → (2,1) (3,1) (1,5)

Expected O/P → (1,5) (2,1) (3,1)

↓

Obs: Data is sorted based on x

① Sort based on distance

`Collections.sort(A, new Comparator<Point> () {`

`@Override`

`public int comp (Point a, Point b) {`

`int dista = (a.x)2 + (a.y)2`

`int distb = (b.x)2 + (b.y)2`

`if (dista > distb)`

`return -1`

`else if (distb > dista)`

`return 1`

`else`

`return 0`

`class Point {`

`int x, y`

`}`

② Copy first B elements of A[] → ans[]

③

Collections.sort(ans, new Comparator<Point> () {

@Override

public int comp (Point a, Point b) {

if (a.x < b.x)

return -1

else if (b.x < a.x)

return 1

else {

if (a.y < b.y)

return -1

else

return 1

x

↓

a[0]

y

↓

a[1]

$N \log N \times TC(\text{comparator})$

TC: $O(N \log N)$

SC: $O(\text{sorting algo of language})$
