

Greedy Problem Solving

- Priyansh Agarwal

Doubled Array Problem: [Link](#)

$$N \leq 10^5$$

$$10^{-9} \leq A[i] \leq 10^9$$

There is an array with N elements, each of the elements is doubled and appended into the array. After that the array elements are jumbled up.

Find the original elements in any order.

Example:

$A = [2, 9, 4, 10]$ gets transformed to $[8, 10, 9, 4, 2, 18, 4, 20]$

$A = [-1, 2, 4]$ gets transformed to $[-2, -1, 2, 4, 4, 8]$

Jumbled array :

[2 4 4 8 9 10 18 20]

Assuming array elements are ≥ 0

$$[a_1 \leq a_2 \leq a_3 \leq a_4]$$

[a_1 a_2 a_3 a_4 b_1 b_2 b_3 b_4]

$$[1 \quad 2 \quad 4 \quad 6]$$



$$1 \quad 2 \quad 4 \quad 6 \quad 2 \quad 4 \quad 8 \quad 12$$

Diagram illustrating a sequence of numbers with arcs connecting them, suggesting a mapping or transformation. Arcs connect 1 to 2, 2 to 4, 4 to 6, 6 to 2, 2 to 4, 4 to 8, and 8 to 12. An upward arrow points to the number 12.

$$a_1 \leq a_2 \leq a_3 \dots \leq a_n \leftarrow$$

$$2a_1 \leq 2a_2 \leq 2a_3 \dots \leq \boxed{2a_n}$$

$[a_1 \quad a_2 \quad a_3 \quad b_1 \quad b_2 \quad b_3 \quad a_4 \quad \boxed{b_4}]$

$$a_1 \leq a_2 \leq a_3 \leq a_4$$

$\lfloor b_4/2 \rfloor$ (highest value in doubled
array cannot be present in the
original array)



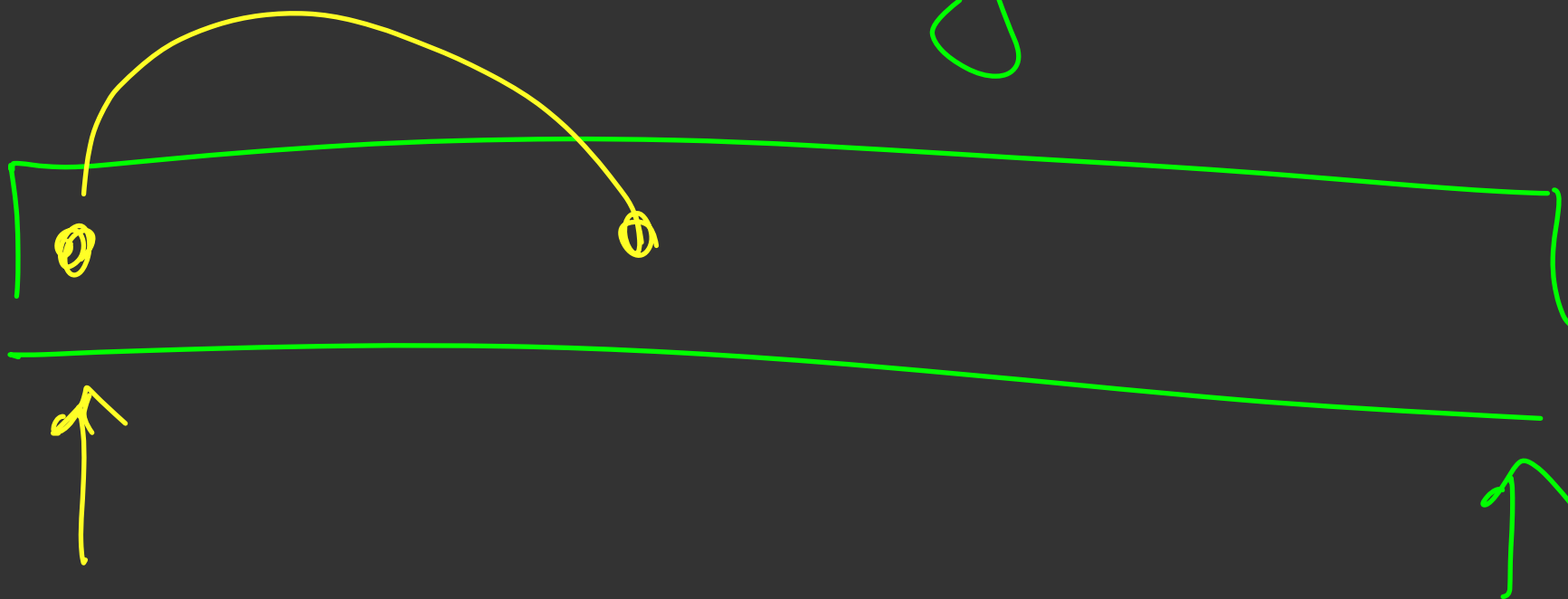
$$a_1 \leq a_2 \leq \delta_1 \leq a_3 \leq b_2 \leq \delta_3 \leq a_4 \leq \delta_4$$

The above inequality chain is enclosed in a large green box. Green checkmarks are placed above the first three terms (a_1, a_2, δ_1), and green arrows point to the terms $a_3, \delta_3, a_4,$ and δ_4 .

Initial problem $\rightarrow \{a_1, a_2, a_3, a_4\}$

New problem $\rightarrow \{a_1, a_2, a_3\}$

doubled array



$[0 \ 0 \ 1 \ 2 \ 3]$

$[0 \ 0 \ 0 \ 0]$ $[1 \ 2 \ 3 \ 2 \ 4 \ 6]$

$$[-1 \quad -2 \quad -4] \quad 0$$

$$[-1 \quad -2 \quad -4 \quad -2 \quad -4 \quad -8]$$

$$[-8 \quad -4 \quad -4 \quad -2 \quad -2 \quad -1]$$

$$0 \leq a_1$$

$$a_1 < 0$$

$$2a_1 \geq a_1$$

$$2a_1 < a_1$$



multiset m;

for (int i = 0; i < 2n; i++) {

m.insert(a[i]);

} bool ans = true

while (m.size() > 0) {

int highest = ~~1~~ prev(m.end());

if (highest > 0)

m.erase(m.find(highest/2));

m.erase(m.find(highest*2));

check
to remove

else

m.erase(m.find(highest));

y

Doubled Array Problem

Can we separate out the negative and positive numbers? ✓

What about the smallest positive number and biggest negative number?

How to simulate this process to get the answer at every step?

Problem ²~~3~~: Link

$$a_1 \quad a_2 \quad \dots \quad a_n$$

↓

$$2^{a_1} \quad 2^{a_2} \quad \dots \quad 2^{a_n}$$

$$2^{a_1} \quad 2^{a_2} \quad 2^{a_4}$$

$$\begin{array}{cccc} a_5 & a_6 & a_7 & a_8 \\ 2 & 2 & 2 & 2 \end{array}$$

1st observation :

If in one step we are including
 $\underline{a_1} \quad \underline{a_2} \quad \dots \quad \underline{a_k}$ such that
 $a_1 \leq a_2 \leq \dots \leq a_k$ then

a_1 cannot just occur once

$$a_1 = 1$$
$$\underline{\underline{2^1}}$$

$$a_2 = 2$$
$$\underline{\underline{2^3}}$$

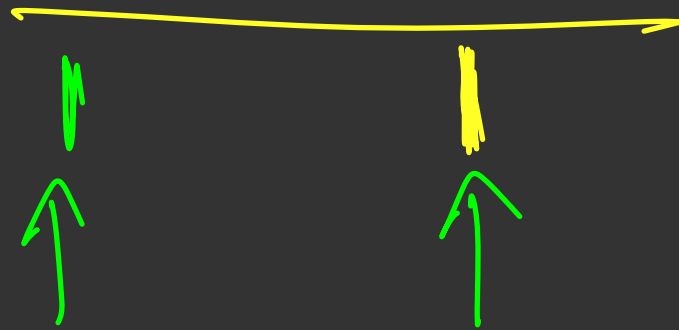
$$a_2 = 4$$
$$\underline{\underline{2^4}}$$

10 ↓ 1050 10000

~~~~~ 10

1000

10000



2nd operation :

Can two smallest number occur  
3 times



$$\begin{array}{r}
 100 \\
 100 \\
 100 \\
 2000 \\
 2000 \\
 \hline
 \end{array}$$

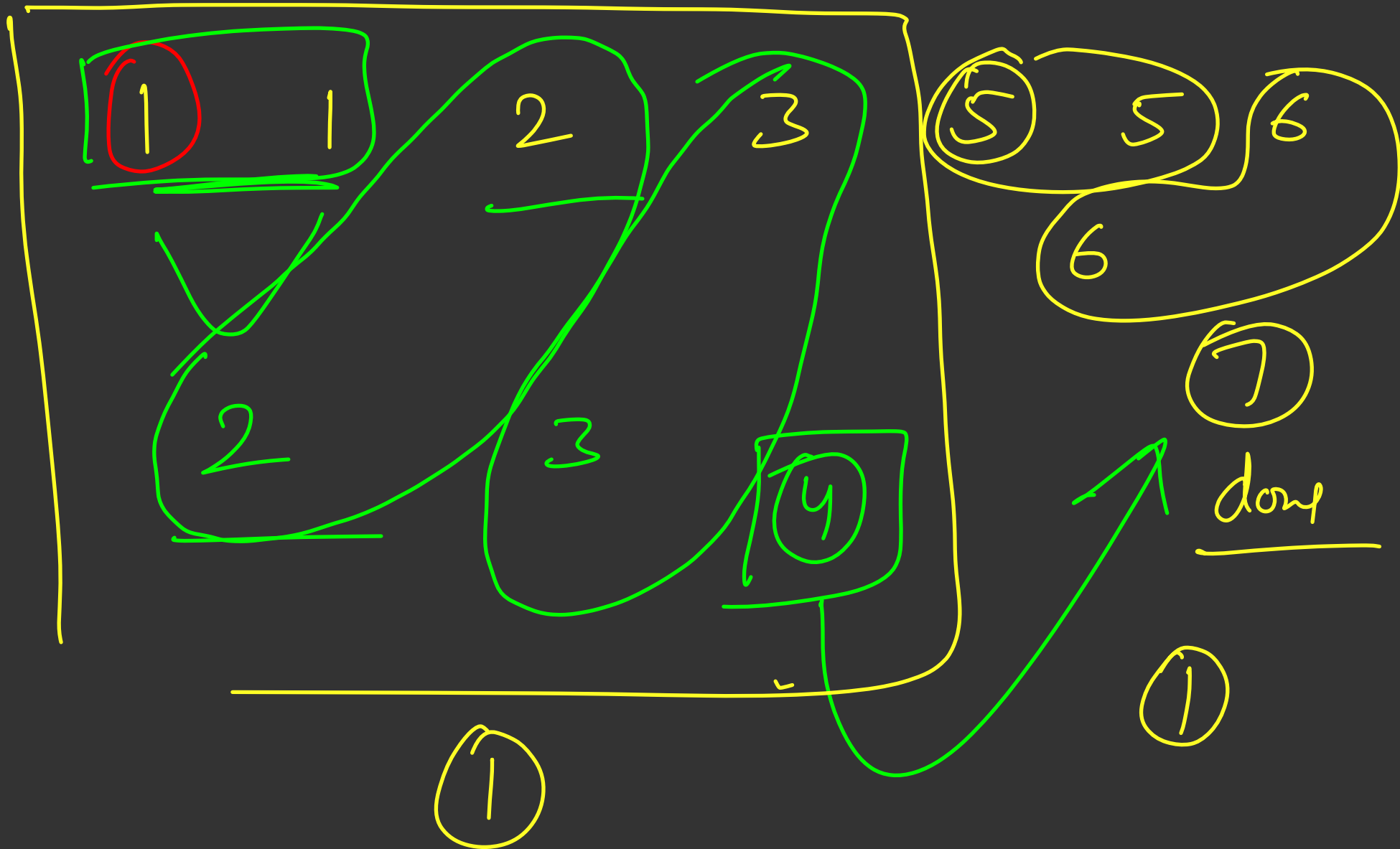
↓

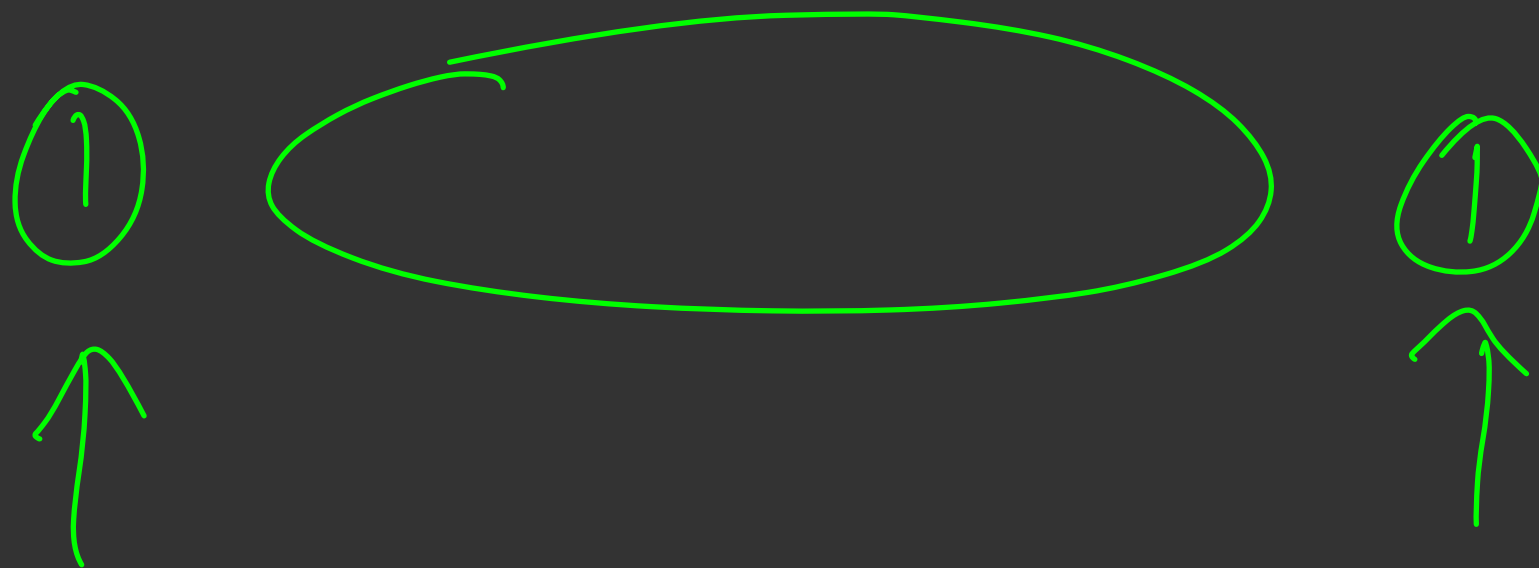
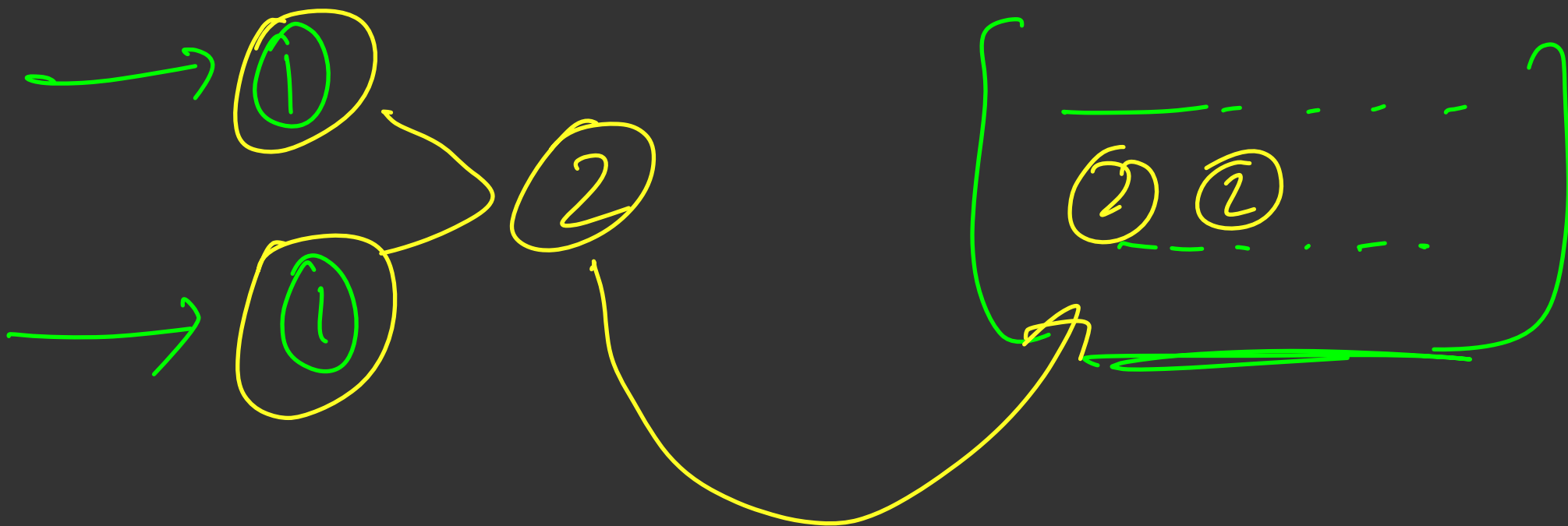
~~100~~

$$2^3$$

# if in a set we have more  
than 1 element then the  
smallest element must occur  
even no of times

# if it contains 1 element then  
it is true





$$\left[ \begin{array}{cc|cccc} 1 & 1 & >1 & >1 & >1 & >1 & \dots \end{array} \right]$$

$$\begin{array}{l} 1 \\ 1 \end{array} \quad \begin{array}{l} \diagdown \\ \diagdown \end{array} \quad \begin{array}{l} 2^1 + 2^1 \rightarrow \boxed{2^2} \end{array}$$

$$\left[ >1 >1 >1 >1 \dots \right]$$

Don't clear

multiset m;

for (int i = 0; i < n; i++) {

    m.insert(a[i]);

} int ans = 0

while (m.size() > 0) {

    int x = \*m.begin();

    m.erase(m.find(x));

    if (m.find(x) == m.end())

        m.erase(m.find(x));

        m.insert(x+1);

    else

1 1 1 2 3

5 5 6

↓ 1

1

2 2 3

3 3

ans = 1

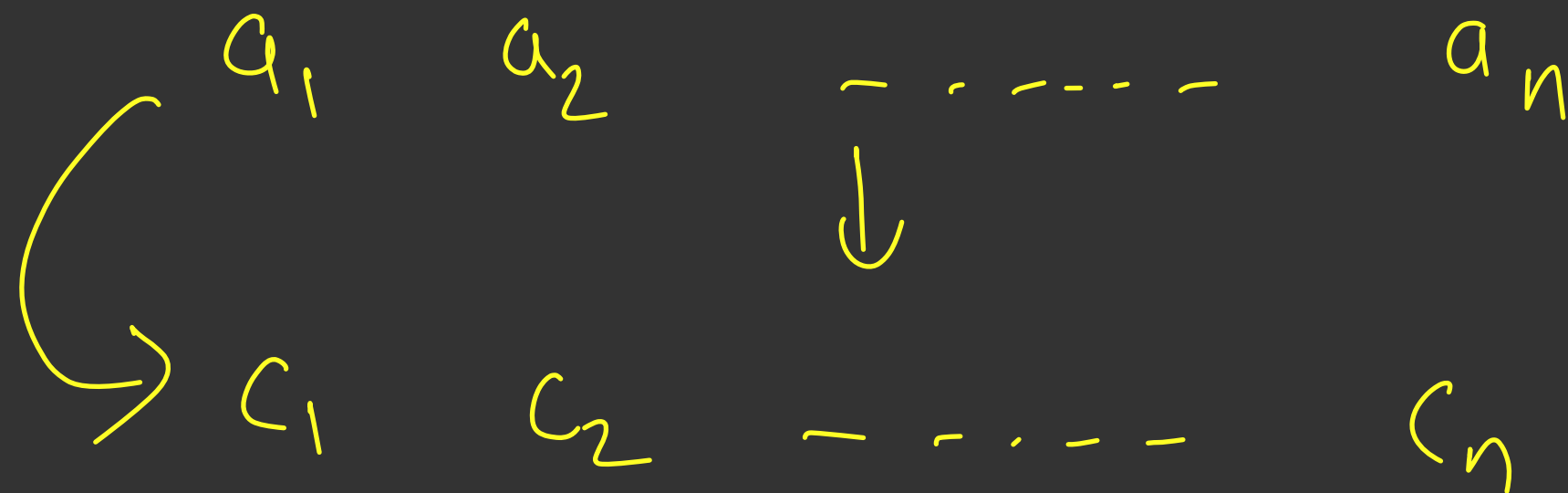
4

L y

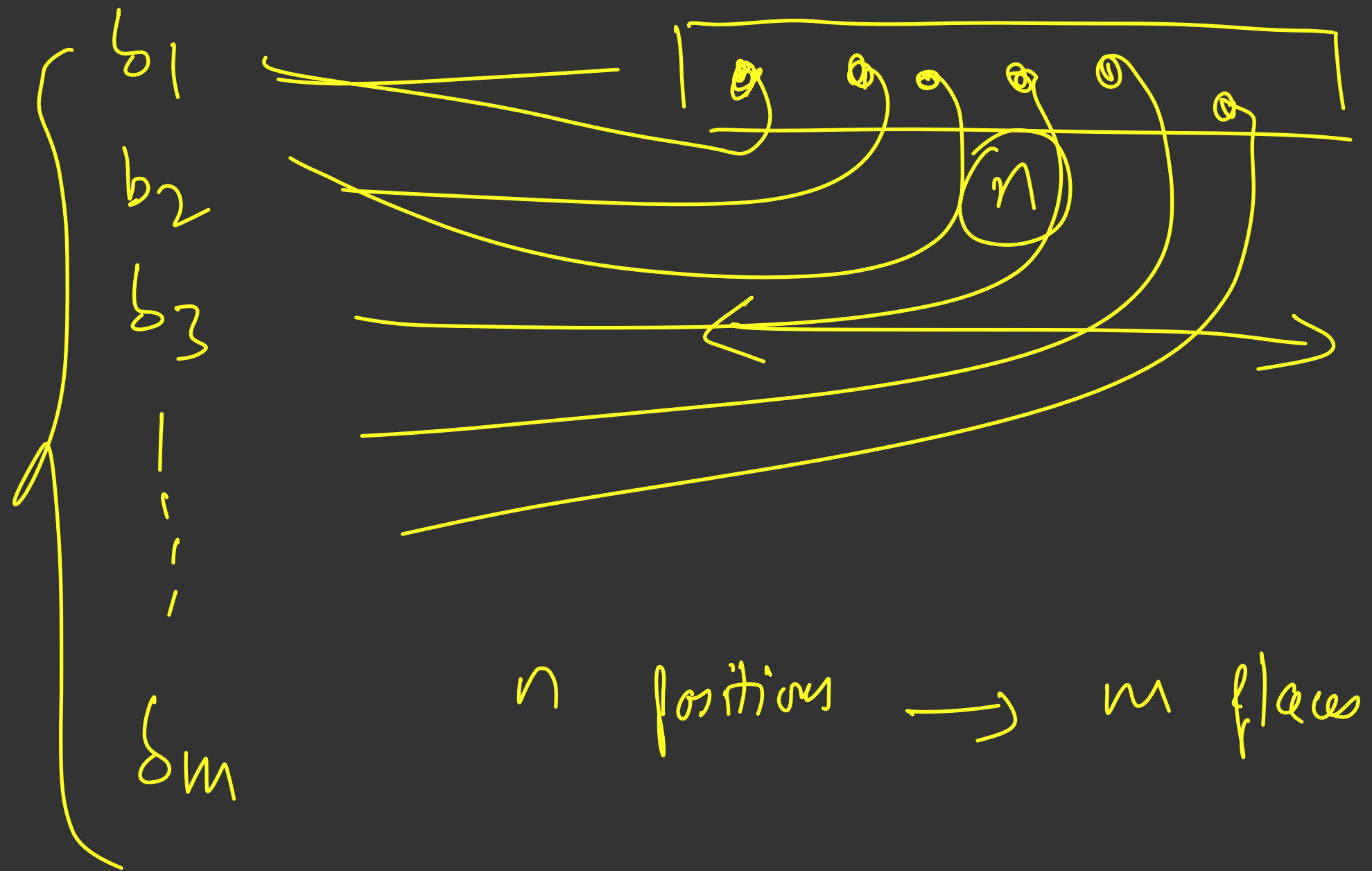
aws + t



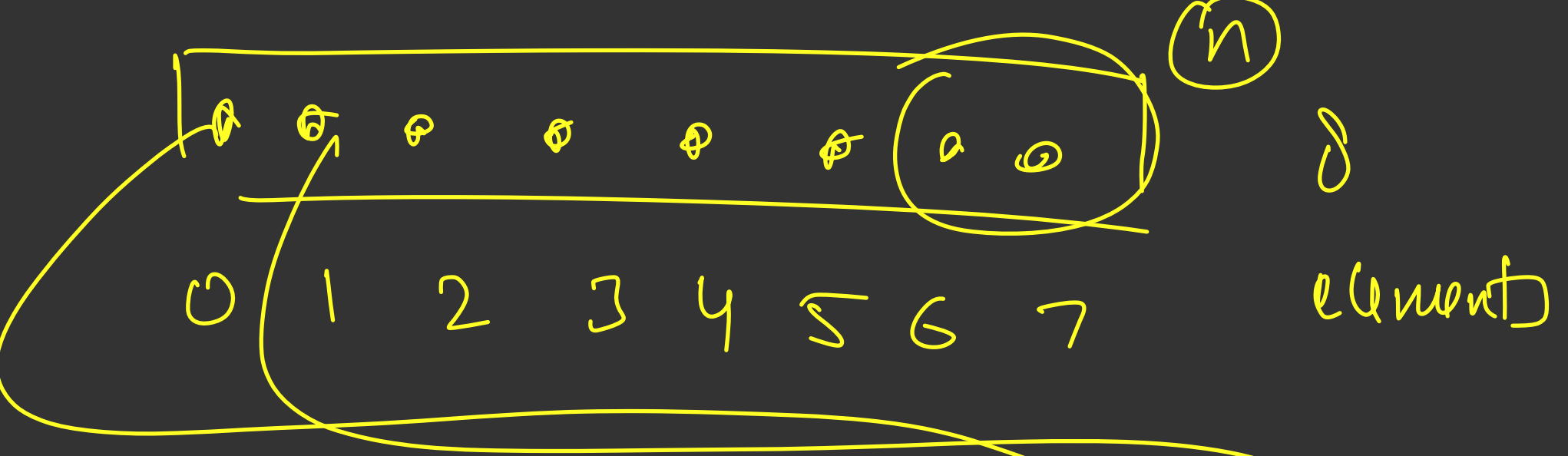
Problem <sup>3</sup>~~4~~: Link



maximize the minimum value in B

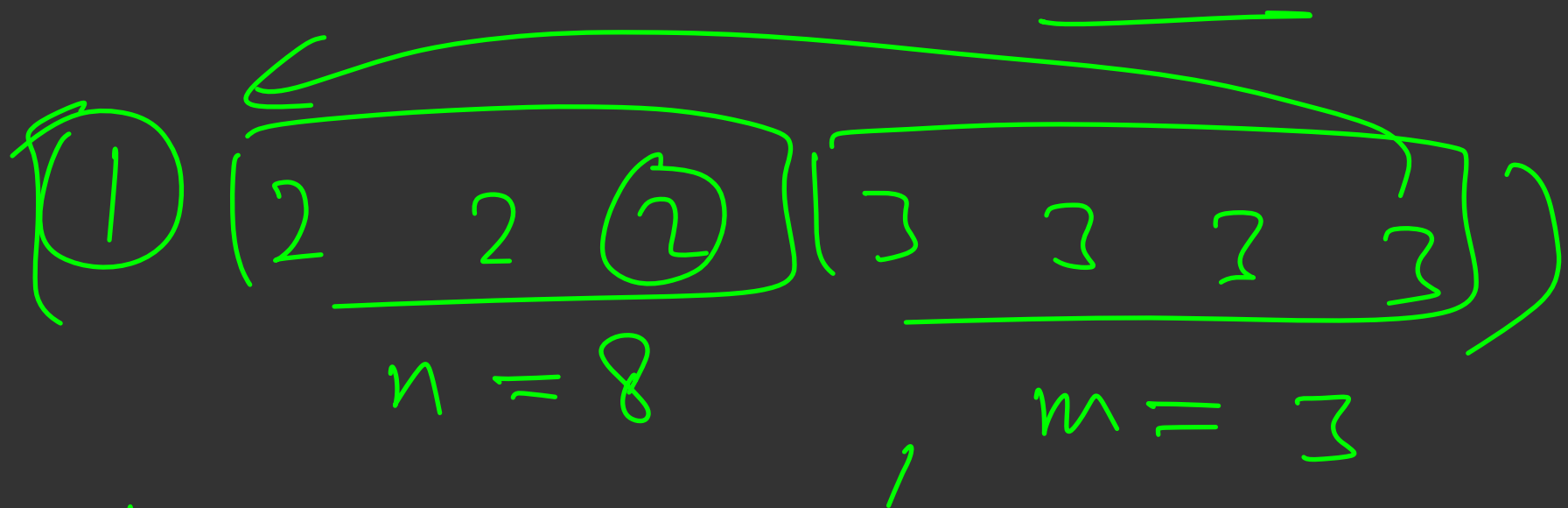


$n$  positions  $\rightarrow$   $m$  places



$$\begin{aligned}
 m &= 3 & b_1 &\leftarrow n/m & 2 \\
 &= & b_2 && n/m & 2 \\
 8/3 &\neq 2 & b_3 && n/m & 2
 \end{aligned}$$

maximum possible (minimum value  
in array B) =  $n/m$



$$n/m = 8/3 = \textcircled{2}$$

# of steps

$$= \sum_{i=1}^m \max(0, n/m - \text{freq}[i])$$

~~\_\_\_\_\_~~

1 2 2 2 (3) 3 3 3

