# Lab 4

# Lab4 A: Block

- Given a sequence of length $n$ , denoted by $a_1, a_2, a_3, \ldots\ldots, a_n$. Each step you can choose an interval $[\,l, r\,]$, $(1 \leqq l \leqq r \leqq n)$, such that all elements within this interval are either incremented by one or decremented by one.
- The task is to find out the minimal steps to make all the numbers in the sequence same.

# Sample Input:

```
3
```
Testcase number

```
6
4 2 0 1 4 3
```
case 1

```
9
18 27 99 2 40 87 11 11 50
```
case 2

```
11
42 16 40 52 14 34 69 81 47 31 65
```
case 4

# Testcase 1

| | l | r | operation | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|---|---|
| | | | | 4 | 2 | 0 | 1 | 4 | 3 |
| Step 1 | 3 | 4 | increment | 4 | 2 | 1 | 2 | 4 | 3 |
| Step 2 | 3 | 3 | increment | 4 | 2 | 2 | 2 | 4 | 3 |
| Step 3 | 2 | 4 | increment | 4 | 3 | 3 | 3 | 4 | 3 |
| Step 4 | 2 | 4 | increment | 4 | 4 | 4 | 4 | 4 | 3 |
| Step 5 | 6 | 6 | increment | 4 | 4 | 4 | 4 | 4 | 4 |

# Testcase 1:

| | l | r | operation | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|---|---|
| | | | | 4 | 2 | 0 | 1 | 4 | 3 |
| Step 1 | 5 | 6 | decrement | 4 | 2 | 0 | 1 | 3 | 2 |
| Step 2 | 5 | 6 | decrement | 4 | 2 | 0 | 1 | 2 | 1 |
| Step 3 | 1 | 1 | decrement | 3 | 2 | 0 | 1 | 2 | 1 |
| Step 4 | 1 | 1 | decrement | 2 | 2 | 0 | 1 | 2 | 1 |
| Step 5 | 3 | 4 | increment | 2 | 2 | 1 | 2 | 2 | 1 |
| Step 6 | 3 | 3 | increment | 2 | 2 | 2 | 2 | 2 | 1 |
| Step 7 | 6 | 6 | increment | 2 | 2 | 2 | 2 | 2 | 2 |

## Analysis of Lab4A

```
6
4 4 4 4 4 4
```

```
6
2 2 2 2 2 2
```

```
6
3 2 2 2 2 2
```

```
6
3 3 3 3 3 4
```

```
6
3 2 2 2 2 1
```

```
6
2 2 4 4 2 2
```

## Analysis of Lab4A

1. **We should choose an interval $[l, r]$ with the same value; how do we find an interval where each element has the same value?**
2. **If multiple intervals satisfy the above requirement, which one should be chosen first?**

## Analysis of Lab4A

1. **We should choose an interval $[\,l, r]$ where each element has the same value; how can we identify such an interval?**

**Let** $\boldsymbol{diff}[\boldsymbol{i}] = \begin{cases} \boldsymbol{a}[1], \ \boldsymbol{where\ i = 1} \\ \boldsymbol{a}[\boldsymbol{i}] - \boldsymbol{a}[\boldsymbol{i} - 1], \ \boldsymbol{where\ i \geq 2} \end{cases}$

```
   a: 4 4 4 4 4 4
diff: 4 0 0 0 0 0
```

```
   a: 2 2 2 2 2 2
diff: 2 0 0 0 0 0
```

```
   a: 3  2 2 2 2 2
diff: 3 -1 0 0 0 0
```

```
   a: 3 3 3 3 3 4
diff: 3 0 0 0 0 1
```

```
   a: 3  2 2 2 2  1
diff: 3 -1 0 0 0 -1
```

```
   a: 2 2 4 4  2 2
diff: 2 0 2 0 -2 0
```

**If in an interval $[i, j]$ (where I > 1) in $\boldsymbol{diff}$, each element is 0, it implies the interval $[\boldsymbol{i} - 1, \boldsymbol{j}]$ in the sequence $\boldsymbol{a}$ satisfies each element has the same value.**

## Analysis of Lab4A

```
   a: 4 4 4 4 4 4
diff: 4 0 0 0 0 0
```

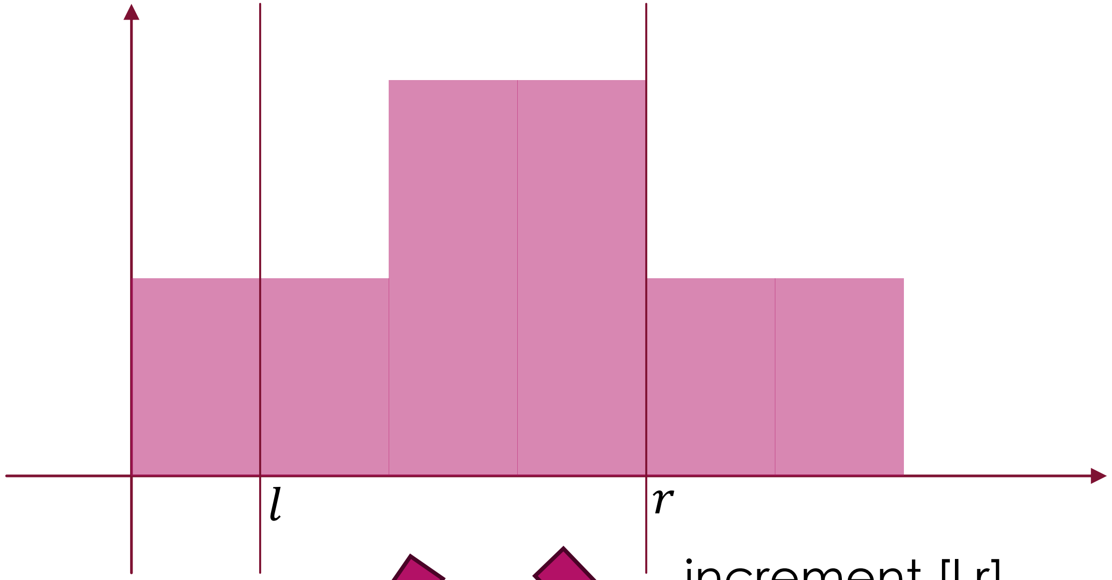
```
   a: 2 2 2 2 2 2
diff: 2 0 0 0 0 0
```

$$\boldsymbol{a[1] = a[2] = a[3] = \cdots = a[n]}$$

$$\Updownarrow$$

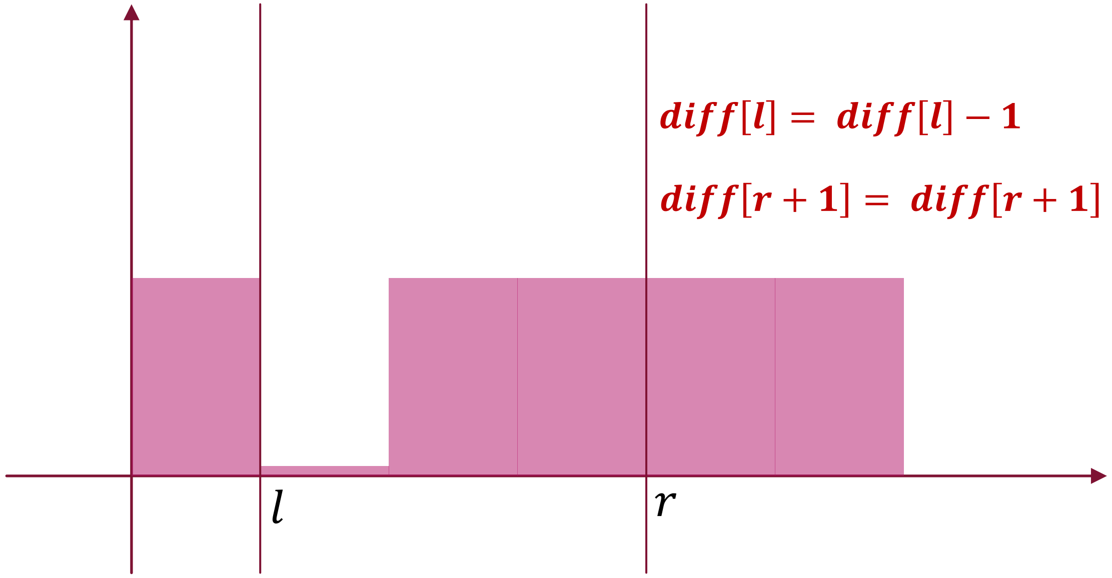
$$\boldsymbol{diff[2] = diff[3] = \cdots = diff[n] = 0}$$

## Analysis of Lab4A



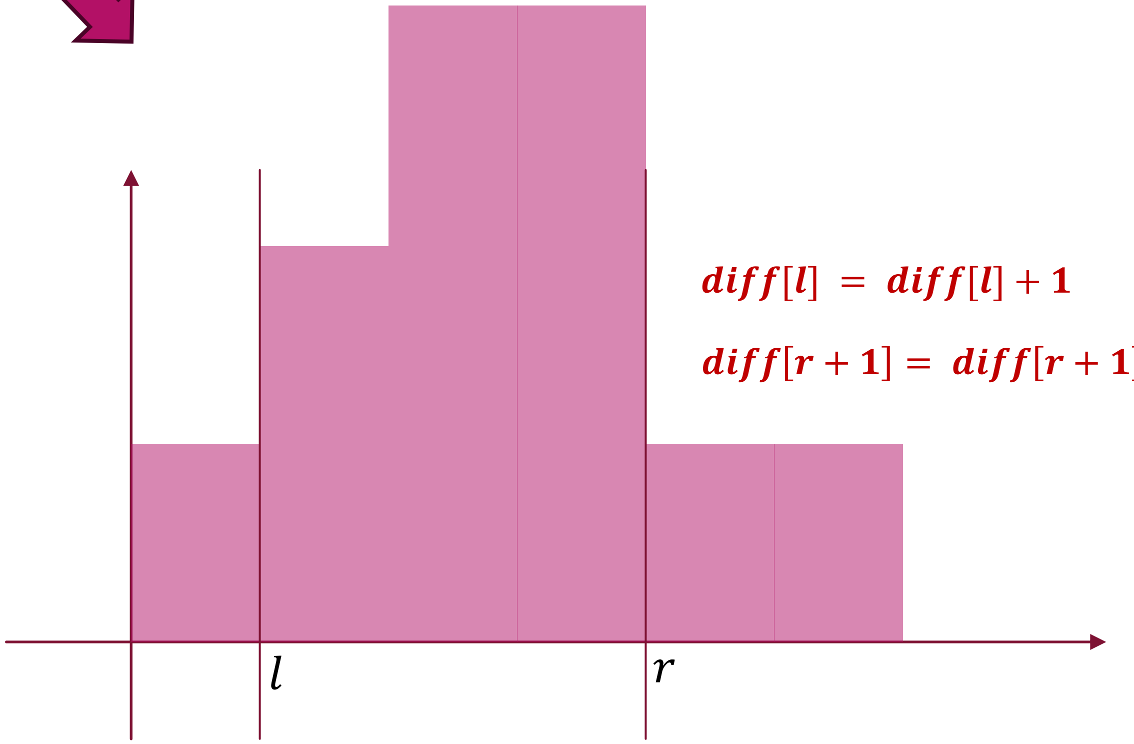

increment [l,r]

decrement [l,r]




$$diff[l] = diff[l] - 1$$

$$diff[r+1] = diff[r+1] + 1$$




$$diff[l] = diff[l] + 1$$

$$diff[r+1] = diff[r+1] - 1$$

## Analysis of Lab4A

```
a:    3   2  2  2  2  2
diff: 3  -1  0  0  0  0
```

**decrement[1,1]**

```
a:    2   2  2  2  2  2
diff: 2   0  0  0  0  0
```

diff[1] = diff[1]-1
diff[2] = diff[2]+1

**increment[1,1]**

```
a:    4   2  2  2  2  2
diff: 4  -2  0  0  0  0
```

diff[1] = diff[1]+1
diff[2] = diff[2]-1

**decrement[2,6]**

```
a:    3   1  1  1  1  1
diff: 3  -2  0  0  0  0
```

diff[2] = diff[2]-1

**increment[2,6]**

```
a:    3   3  3  3  3  3
diff: 3   0  0  0  0  0
```

diff[2] = diff[2]+1

**decrement[1,1] or increment[2,6] can make the diff[2] + 1, then diff[2] become 0**

## Analysis of Lab4A

```
   a:  3  3  3  3  3  4
diff:  3  0  0  0  0  1
```

**decrement[6,6]**

```
   a:  3  3  3  3  3  3
diff:  3  0  0  0  0  0
```

diff[6] = diff[6]-1

**increment[6,6]**

```
   a:  3  3  3  3  3  5
diff:  3  0  0  0  0  2
```

diff[6] = diff[6]+1

**decrement[1,5]**

```
   a:  2  2  2  2  2  4
diff:  2  0  0  0  0  2
```

diff[1] = diff[1]-1
diff[6] = diff[6]+1

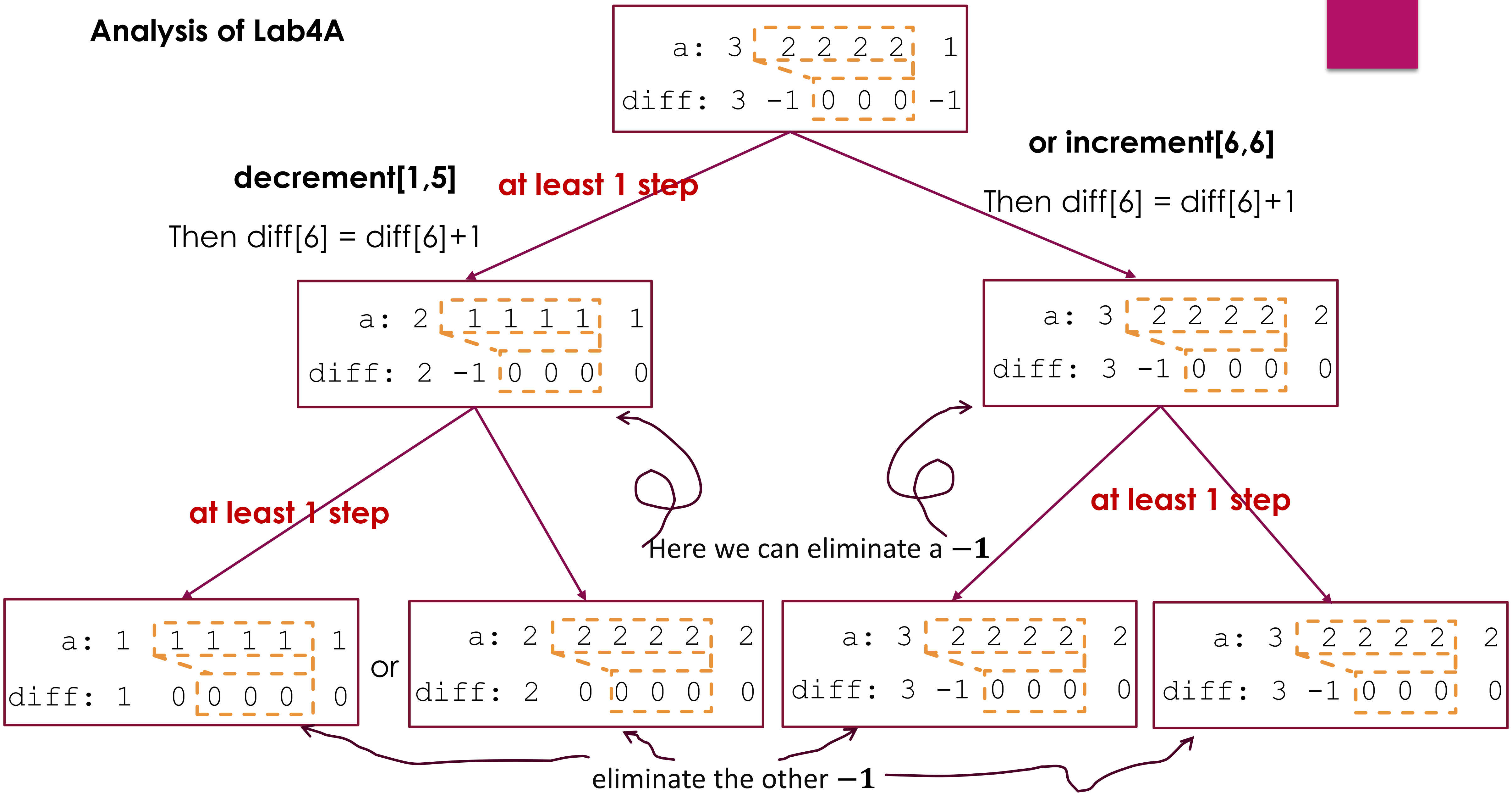
**increment[1,5]**

```
   a:  4  4  4  4  4  4
diff:  4  0  0  0  0  0
```

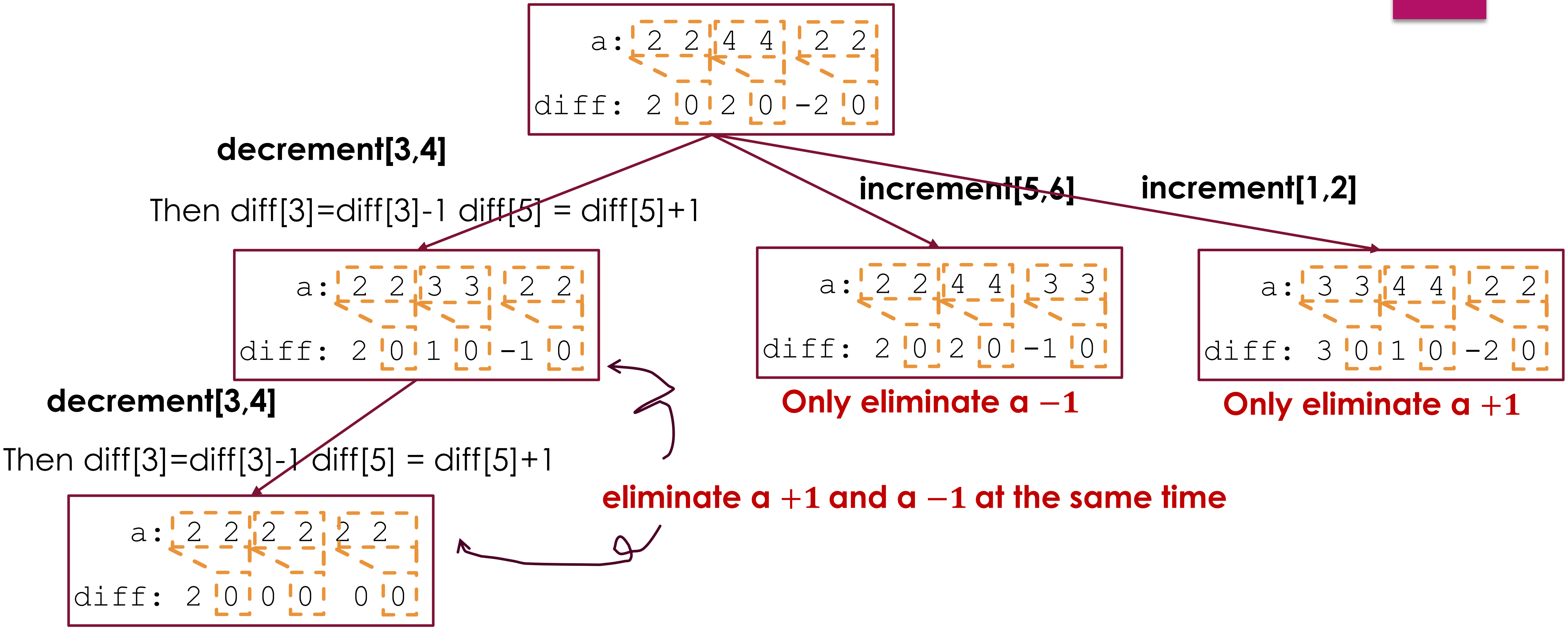
diff[1] = diff[1]+1
diff[6] = diff[6]-1

**decrement[6,6] or increment[1,5] can make the diff[6] - 1, then diff[6] become 0**

# Analysis of Lab4A



## Analysis of Lab4A



## Solution of Lab4A

After constructing array $diff$, we can greedily select pairs of numbers $diff$[i] and $diff$[j] with opposite signs each time.
Assume that the sum of positive numbers are $p$ and the sum of negative numbers are $q$.

**The elimination of one positive and one negative number requires at least min(p, |q|) operations, and for the remaining part |p - |q||, it requires at least |p - |q|| operations.**

```
6
4   2  0 1 4  3
```

**case 1**

```
diff: 4  -2 -2 1 3 -1
```

| $p$ | $q$ |
|---|---|
| 4 | 5 |

$$\min(p, |q|) + \big|p - |q|\big| = \min(4,5) + |4 - 5| = 5$$

# Lab4.B: Profit

- Given a sequence $a_1, a_2, a_3, \ldots\ldots, a_n$. The task is to select at most k numbers from the sequence with the constraint, and output the maximum possible sum of the selected numbers. The constraint is that no two selected numbers can be adjacent.

# Sample Input:

8 3

1 6 9 -6 2 1 3 1

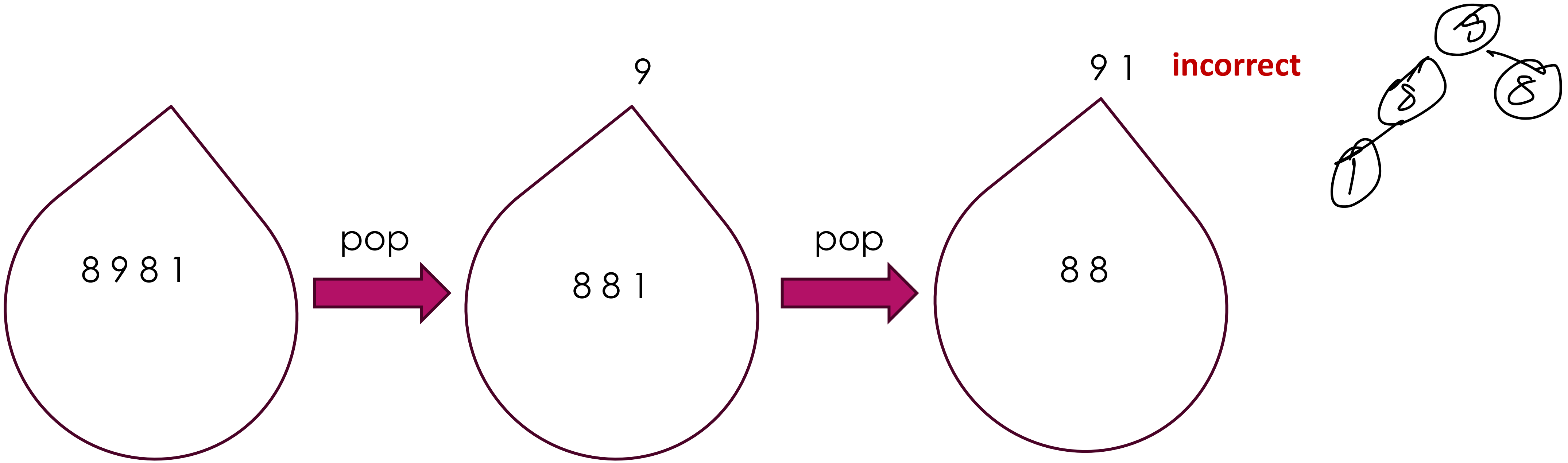
# Sample Output:

**14**

1 6 **9** -6 **2** 1 **3** 1

## Analysis of Lab4B

Using a max heap to ensure that the elements at the top of the heap are not adjacent to each other is an **incorrect approach**.




**8** 9 **8** 1

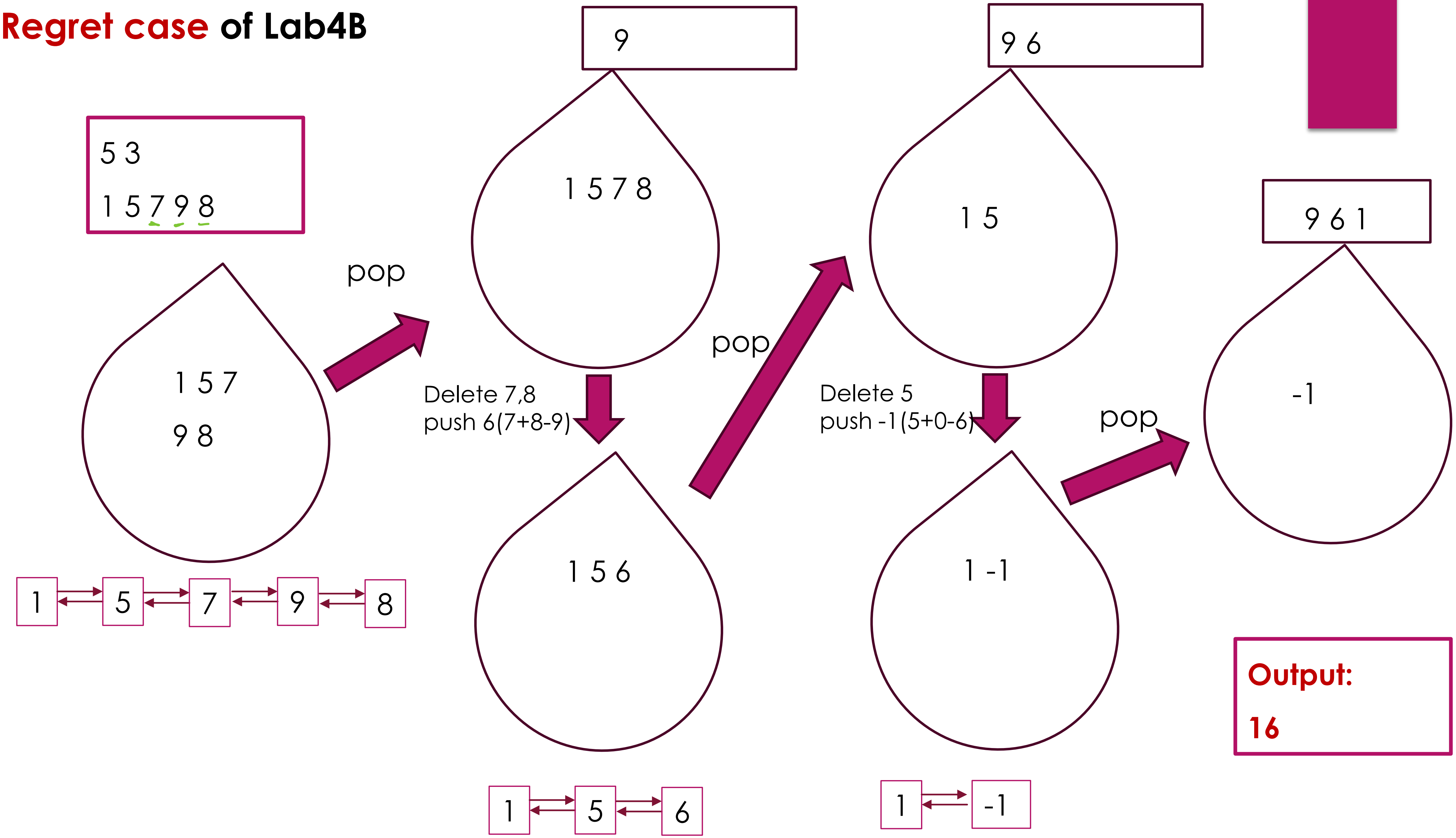
## Analysis of Lab4B

We can improve this process by considering the potential need for **regret** after greedily choosing the largest number and placing **the cost of regret** into the heap.

**Assume the current maximum value is num[i],**

**the cost of regret = num[i+1] + num[i-1] - num[i].**

# Regret case of Lab4B



# Regret then regret case of Lab4B