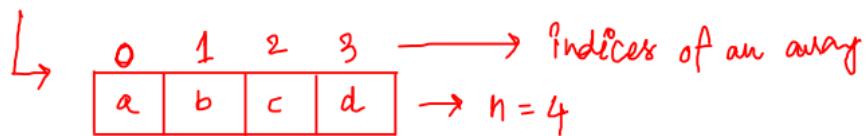


* Arrays:

→ Data structures are used to store/organize and retrieve/access the data efficiently.

① `const friends = ["a", "b", "c", "d"] ;`



diagrammatic / memory representation

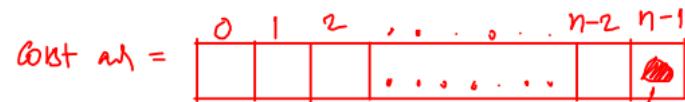
② Accessing array elements through index,

`friends[0]` → "a"

`friends[2]` → "c"

③ `friends.length` → 4

④ If there are n elements in the array, get the last element?



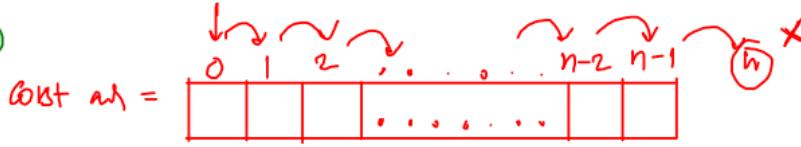
6: `friends[n-1]`

`friends[4-1]`

`friends[3] → "d"`

`arr[n-1]`

⑤



c1(arr[0])

c1(arr[1])

.

:

.

c1(arr[n-2])

c1(arr[n-1])



```
for (let i=0; i<n; i++) {
    console.log(arr[i]);
}
```

 $i \leq n-1$

(n)

* Iterating on
every index

① $i=0 \Rightarrow arr[0]$ ② $i=1 \Rightarrow arr[1]$

:

:

:

③ $i=n-1 \Rightarrow arr[n-1]$ ④ $i=n \Rightarrow terminate$

⑥ changing elements

const arr =

0	1	2	3
a	b	c	d

e



arr[1] = 'e'

0	1	2	3
a	e	c	d

#Qs: You said, we can't change value of a const variable but here we are changing, how?

A: we are not changing value of arr,
we are changing value of arr [1];
const arr = ['a', 'b', 'c', 'd'];
arr = 10; X (this is against const rule)
arr [1] = "e" ✓ (this is not)

⑦ How to add new elements

const arr =

0	1	2	3	4
a	b	c	d	e

* arr.push('e');

8

```

635 // 8) arrays can store any data type
636 function print() {
637   console.log("hello");
638   return 10;
639 }
640
641 const firstName = "Anurag";
642 const age = 23;
643 const job = "Teacher";
644
645 const myArr = [firstName, age, job, friends, print, print()];
646 console.log(myArr);
647 console.log(myArr[0]); -> Anurag
648 console.log(myArr[1]); -> 23
649 console.log(myArr[2]); -> Teacher
650 console.log(myArr[3]);
651 console.log(myArr[4]); -> print def
652 console.log(myArr[4]());
653 console.log(myArr[5]); -> 10
654
655 // given myArr, print the number of friends you have
656 console.log(myArr[3].length);
657
658 // given myArr, print the second friend
659 console.log(myArr[3][1]);
660
661 // given myArr, get the last friend
662 const numFrnds = myArr[3].length;
663 console.log(myArr[3][numFrnds - 1]);

```

myArr[4]();print(); \Rightarrow 10.

function at <686>

0	1	2	3	4	5
"Anurag"	23	"Teacher"	10	10	10

0	1	2	3	4	5
"Roshan"	"Tawn"	"Aqam"	"priya"	"Anjay"	"JavaScript"

myArr[3].length \Rightarrow 6

0	1	2	3	4	5
"Roshan"	"Tawn"	"Aqam"	"priya"	"Anjay"	"JavaScript"

length

0	1	2	3	4	5
"Roshan"	"Tawn"	"Aqam"	"priya"	"Anjay"	"JavaScript"

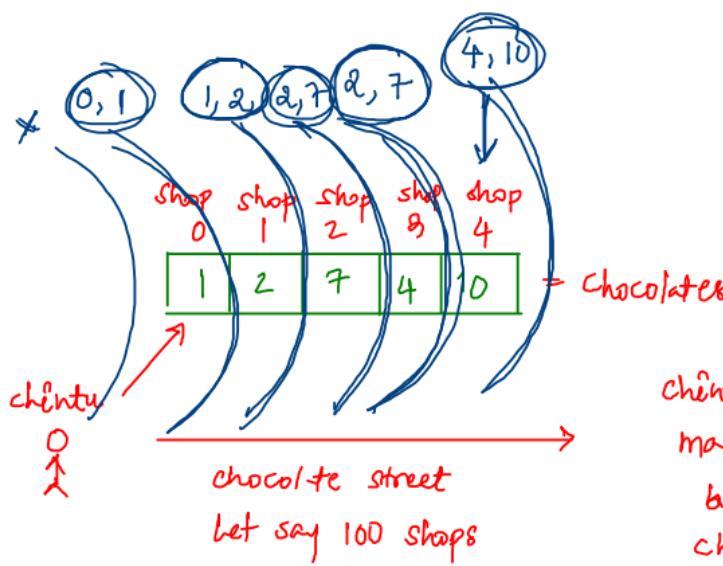
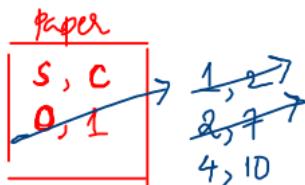
myArr[3][1]

[r-1] \Rightarrow [5] \Rightarrow "JavaScript"

* Max Elt and its Index :

Ex: $[0, 1, 2, 3, 4]$
 $[1, 2, 7, 4, 10]$

Op: 10 4



chintu wants to buy maximum chocolates but he can only choose any one shop!
Help chintu.

- ① go to shop-0 $\Rightarrow s=0, c=1$
- ② go to shop-1 $\Rightarrow s=1, c=2$ Is this shop better than paper? ✓
Correct choc > paper choc
- ③ go to shop-2 $\Rightarrow s=2, c=7$ Is this shop better? ✓
- ④ go to shop-3 $\Rightarrow s=3, c=4$ Is this shop better? ✗
- ⑤ go to shop-4 $\Rightarrow s=4, c=10$ Is this shop better? ✓

#Code:

let maxElc = $-\infty$; (Initialize with lowest value possible)

let idx(s) = -1; (Initialize with something which is not possible)

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

if (arr[i] > maxElc) {

maxElc = arr[i];

idx = i;

}

5 // If current choc > paperchoco
update paper

[$\begin{smallmatrix} 0 & 1 & 2 & 3 & 4 \\ 1, 2, 7, 4, 10 \end{smallmatrix}$]

maxElc = $\cancel{-\infty} \cancel{X} \cancel{X} 10$
idx = $\cancel{1} \cancel{X} \cancel{X} 4$

① i=0, arr[0] > $-\infty$, 1 > $-\infty$ ✓

② i=1, arr[1] > 1, 2 > 1 ✓

③ i=2, arr[2] > 2, 7 > 2 ✓

④ i=3, arr[3] > 7, 4 > 7 X

⑤ i=4, arr[4] > 7, 10 > 7

⇒ op: maxElc idx

10 4

Q3: can we take maxEle = 0 (Initialize) [This only works for +ve numbers]

i	0	1	2	3
	-1	-2	-3	-4

[If array has -ve numbers also take $-\infty$]

maxEle = 0

- ① $i=0$, $\text{arr}[0] > \text{maxEle}$, $-1 > 0 \times$
- ② $i=1$, $\text{arr}[1] > \text{maxEle}$, $-2 > 0 \times$
- ③ $i=2$, $\text{arr}[2] > \text{maxEle}$, $-3 > 0 \times$
- ④ $i=3$, $\text{arr}[3] > \text{maxEle}$, $-4 > 0 \times$

Op: $\text{maxEle} \Rightarrow 0$ (wrong answer)

maxEle = $-\text{Infinity} (-\infty)$

- ① $i=0$, $\text{arr}[0] > \text{maxEle}$, $-1 > -\infty \checkmark$
 $\rightarrow \text{maxEle} = \text{arr}[0] = -1$
- ② $i=1$, $\text{arr}[1] > \text{maxEle}$, $-2 > -1 \times$
- ③ $i=2$, $\text{arr}[2] > \text{maxEle}$, $-3 > -1 \times$
- ④ $i=3$, $\text{arr}[3] > \text{maxEle}$, $-4 > -1 \times$

Op: $\text{maxEle} \Rightarrow -1$ (correct)

Qs: Generating pairs / triplets and so on . . .

eg: $[\begin{smallmatrix} 0 & 1 & 2 & 3 \\ 20, 10, 30, 40 \end{smallmatrix}]$

Op: $\begin{cases} (0, 1) \\ (20, 10) \end{cases}$

① $\begin{cases} (0, 2) \\ (20, 30) \\ (0, 3) \\ (20, 40) \end{cases}$ * no. of pairs = $\frac{n(n-1)}{2} = \frac{4 \times 3}{2} = 6$

② $\begin{cases} (1, 2) \\ (10, 30) \\ (1, 3) \\ (10, 40) \end{cases}$

③ $\begin{cases} (2, 3) \\ (30, 40) \end{cases}$

* keep on chaining loops to get triplet / quadruplet so on . . .

$i \leq n-2$

(a)

for (let $i = 0$; $i < n-1$; $i++$) {

 for (let $j = i+1$; $j < n$; $j++$) {

 cl(ar[i], ar[j]);

}

① $i=0$

$\rightarrow j=1 \Rightarrow ar[0], ar[1] \Rightarrow 20, 10$

$\rightarrow j=2 \Rightarrow ar[0], ar[2] \Rightarrow 20, 30$

$\rightarrow j=3 \Rightarrow ar[0], ar[3] \Rightarrow 20, 40$

② $i=1$

$\rightarrow j=2 \Rightarrow ar[1], ar[2] \Rightarrow 10, 30$

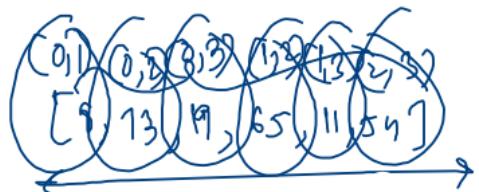
$\rightarrow j=3 \Rightarrow ar[1], ar[3] \Rightarrow 10, 40$

* Max difference b/w any two elements :

e.g.: $\begin{bmatrix} 16, 24, 89, 95 \end{bmatrix}$

8 \leftarrow $(16, 24)$ $(24, 89) \leftarrow 65$ $(89, 95) \rightarrow 54$
13 \leftarrow $(16, 89)$ $(24, 95) \rightarrow 11$
19 \leftarrow $(16, 95)$

op: 73



$(\text{diff} > \text{maxDiff}) \Rightarrow \text{update}$

Math.abs(-ve) \Rightarrow +ve
Math.abs(+ve) \Rightarrow +ve

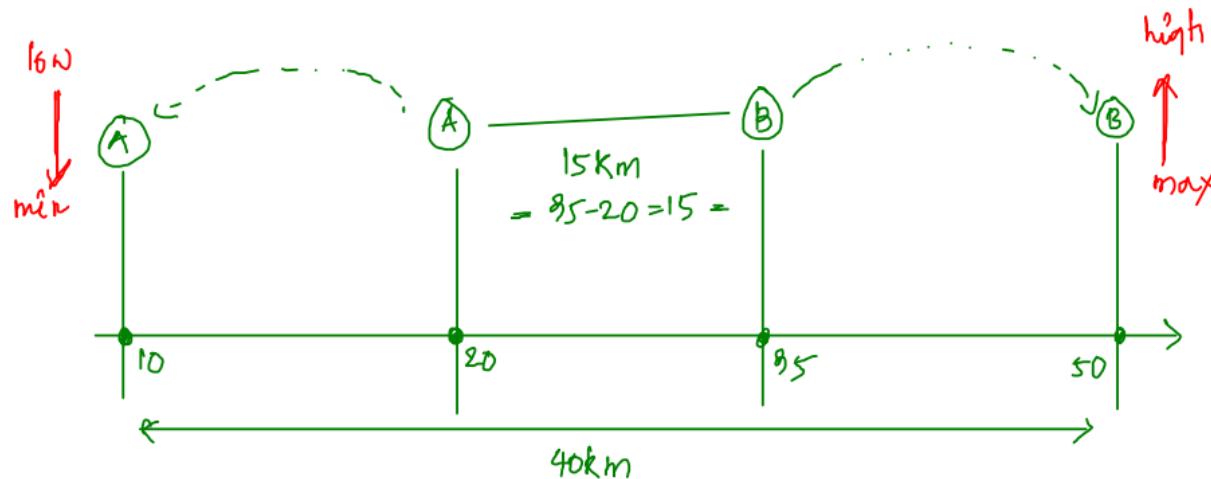
let maxDiff = - ∞ ;

```
for (let i=0; i<n-1; i++) {  
    for (let j=i+1; j<n; j++) {  
        const diff = Math.abs(arr[i]-arr[j]);  
        if (diff > maxDiff) {
```

```
            maxDiff = diff;  
        }  
    }  
}
```

\Rightarrow out of all diff of pairs, pick the maximum possible diff.

Efficient / optimal :



Ans: $\max \text{ele} - \min \text{ele}$

$$\text{Ex: } [16, 24, 39, 50] \rightarrow \begin{array}{l} \max = 50 \\ \min = 16 \end{array} \Rightarrow 50 - 16 = 34$$