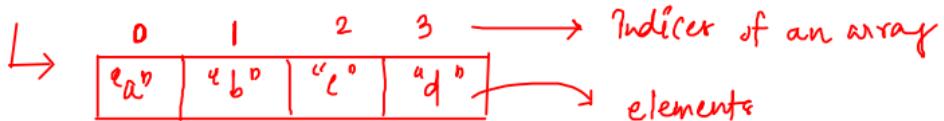


① const friends = ["a", "b", "c", "d"] ; // Created



diagrammatic representation

$$n = 4$$

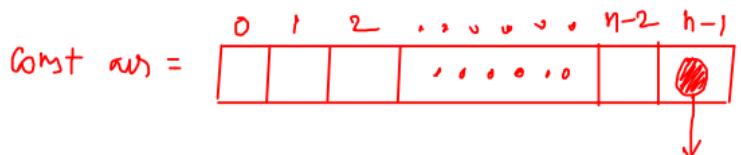
② How to access elements,

friends[0] → "a"

friends[2] → "c"

③ friends.length → 4

④ If there are 'n' elements in the array, what is the last element ?



⇒ friends[n-1]

arr[n-1]

⇒ friends[4-1]

⇒ friends[3] → "d"

⑥ Can I change an element inside an array?

const arr = [
 ⁰a, ¹b, ²c, ³d]

arr[1] = "e"
[
 ⁰a, ~~¹b~~, ²c, ³d]
 ^
 e

* changing
is also called as mutability.

* Are JS arrays mutable? Yes

* we have declared the array as const but we are able to change the element,
How?

→ more about it tomorrow, In simple words,

const arr1 = [~~1, 2, 3, 4~~]
 ^

const arr2 = [5, 6, 7, 8]

X arr1 = arr2

⑦ How to add new elements to the array,

const arr = [
 ⁰a, ¹b, ²c, ³d]

n = 4

arr.push("e")

↓
[
 ⁰a, ¹b, ²c, ³d, ⁴e]

n = 5

⑧ arr.pop(); // to remove last element

⑨

```
747 const myArr = [firstName, age, job, friends];  
748 console.log(myArr);  
749 console.log(myArr[0]); → Amrit  
750 console.log(myArr[1]); → 23  
751 console.log(myArr[2]); → Teacher  
752 console.log(myArr[3]);  
753  
754 // given myArr, find the number of friends  
755 console.log(myArr[3].length);  
756  
757 // given myArr, print the 2nd indexed friend  
758 console.log(myArr[3][2]);  
759  
760 // given myArr, print the last friend  
761 const numFriends = myArr[3].length;  
762 console.log(myArr[3][numFriends - 1]);
```

numFriends = 4

myArr[3][4-1]

= myArr[3][3]

⇒ "Sandip"

0	1	2	3
Amrit	23	Teacher	Sandip

0	1	2	3
"kiran"	"vipin"	"Nikita"	"sandip"

myArr[3].length

0	1	2	3
"kiran"	"vipin"	"Nikita"	"sandip"

⇒ ④

myArr[3][2]

0	1	2	3
"kiran"	"vipin"	"Nikita"	"sandip"

⇒ "Nikita"

* Find the Maximum file and its Index :

Eq:

0	1	2	3	4
1	2	7	4	10

Op: 10, 4

Paper

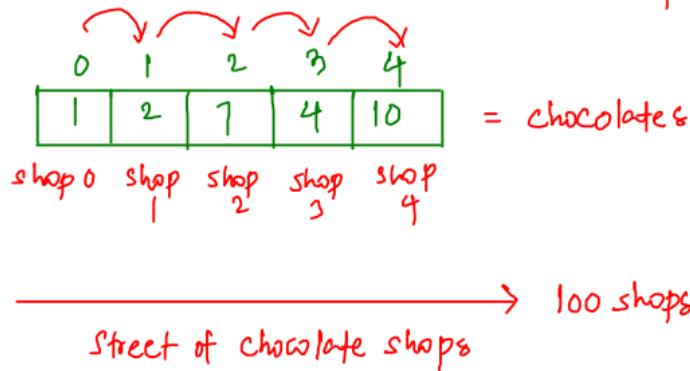
S, C
0, 1
1, 2
2, 7
4, 10

- ① goes to shop 0
- ② goes to shop 1
→ is this a better choice? yes
- ③ goes to shop 2
→ is this better? yes
- ④ goes to shop 3 → is this better? No ($4 < 7$)



1st class
what is Algorithm?
- steps

* kid wants to buy max choc but he can only buy from a single shop



- ⑤ goes to shop 4
→ is this better?
yes

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

$$\left. \begin{array}{l} \text{max Ele} = -\infty \\ \text{max Idx} = -1 \end{array} \right\} \begin{array}{l} \text{initial} \\ \text{values} \end{array}$$

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

// to check whether this shop is better or not?

if ($arr[i] > \text{maxEle}$) {

$\text{maxEle} = arr[i]$

$\text{maxIdx} = i$

}

}

* when you have not seen any shop,

$\boxed{\begin{matrix} \text{shop, choc} \\ -1, -\infty \end{matrix}}$

the shop which doesn't exist

something which is worst

as min as possible
(less expectations)

$\text{maxEle} = -\infty \neq 10$

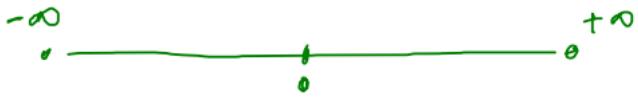
$\text{maxIdx} = -1 \neq 4$

\rightarrow no. of choc in current shop $>$ no. of choc that you have

- ① $i=0$, $arr[0] > \text{maxEle}$, $1 > -\infty$, update
- ② $i=1$, $arr[1] > \text{maxEle}$, $2 > 1$, update
- ③ $i=2$, $arr[2] > \text{maxEle}$, $7 > 2$, update
- ④ $i=3$, $arr[3] > \text{maxEle}$, $4 > 7$, NO
- ⑤ $i=4$, $arr[4] > \text{maxEle}$, $10 > 7$, update

Eg:

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



if $\text{maxfile} = 0$; \Rightarrow take a number which lowest $\rightarrow -\infty$

① $i=0$, and $[0] > \text{maxfile}$, $-4 > 0$, NO

② $i=1$, and $[1] > \text{maxfile}$, $-2 > 0$, NO

③ $i=2$, and $[2] > \text{maxfile}$, $-3 > 0$, NO

④ $i=3$, and $[3] > \text{maxfile}$, $-1 > 0$, NO

\Rightarrow op; max value is ④ X

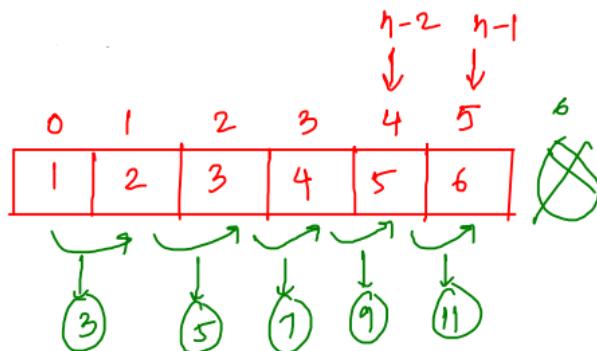


But 0 is not in
the array

* Consecutive Elements Sum :

$$k = 7$$

const arr =



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

 const sum = arr[i] + arr[i+1];

 console.log(sum);

}

* 'i' is consecutive
⇒ 'i+1'

* There is no consecutive ele for 'n-1'

Cnt = 0
1

$$\textcircled{1} \quad i=0 \rightarrow a(0) + a(1)$$

$$\rightarrow 1+2 = \textcircled{3} == k$$

$$\textcircled{2} \quad i=1 \rightarrow a(1) + a(2)$$

$$\rightarrow 2+3 = \textcircled{5} == k$$

$$\textcircled{3} \quad i=2 \rightarrow a(2) + a(3)$$

$$\rightarrow 3+4 = \textcircled{7} == k$$

$$\textcircled{4} \quad i=3 \rightarrow a(3) + a(4)$$

$$\rightarrow 7+9 = \textcircled{16} == k$$

* $\boxed{i=4}$

$$\rightarrow a(4) + a(5)$$

$$\rightarrow 5+6 = \textcircled{11} == k$$

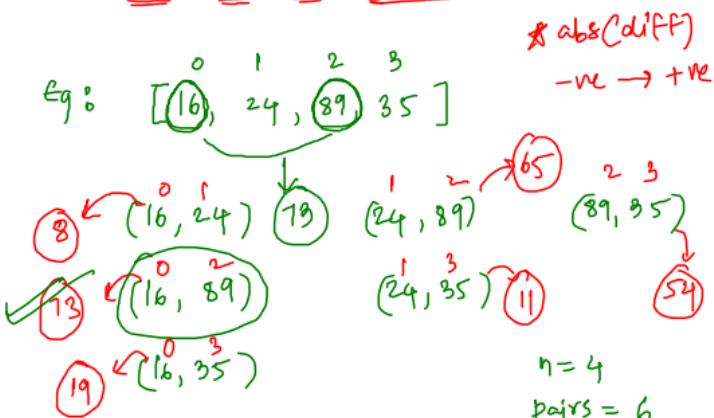
~~$\textcircled{5}$~~

$$\rightarrow a(5) + a(6)$$

$$\rightarrow 6 + \text{undefined} \rightarrow \times$$

This needs to be avoided

* Max diff b/w any 2 elements:



let maxDiff = -Infinity;

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

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

 const diff = Math.abs(arr[i] - arr[j]);

 if (diff > maxDiff) {

 maxDiff = diff;

 }

}

* How to generate any 2 elements?

(0, 1) (1, 2) (2, 3)

(0, 2) (1, 3)

(0, 3)

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

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

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

}

① $i=0 \rightarrow j=1 \times \cancel{\text{P}}$

arr[0], arr[1] $\rightarrow 16, 24$

arr[0], arr[2] $\rightarrow 16, 89$

arr[0], arr[3] $\rightarrow 16, 35$

② $i=1 \rightarrow j=2 \times \cancel{\text{P}}$

arr[1], arr[2] $\rightarrow 24, 89$

③ $i=2 \rightarrow j=3 \times \cancel{\text{P}}$

arr[2], arr[3] $\rightarrow 89, 35$

④ $i=3 \rightarrow j=4 \times \cancel{\text{P}}$

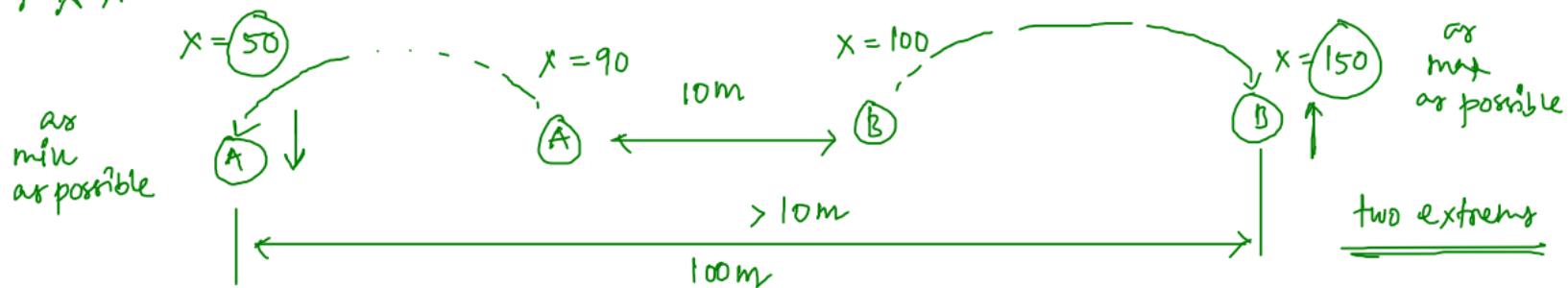
arr[3], arr[4] $\rightarrow 35, 45$

$$\rightarrow \text{How many pairs are possible} = \frac{n(n-1)}{2}$$

$$\frac{\cancel{4} \times 3}{\cancel{2}} = 6 \quad (\text{when } n=4)$$

To generate all pairs, iterations = $\frac{n(n-1)}{2} = \frac{n^2(n-1)}{2} \approx n^2$ iterations

highest impact[ly] term



two elements such that their difference is maximum,

$\Rightarrow \text{maxEle} - \text{minEle}$