Cricket

INTEGER

20, 45, 102, 151, 0, 0 .. string 79

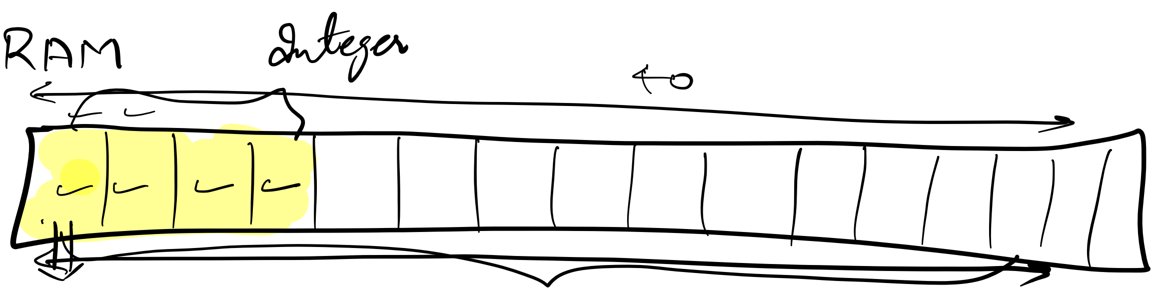
collection

List (Linear)

Arrays

⇒ Array is a collection of same type of data stored at contiguous memory locations

RAM



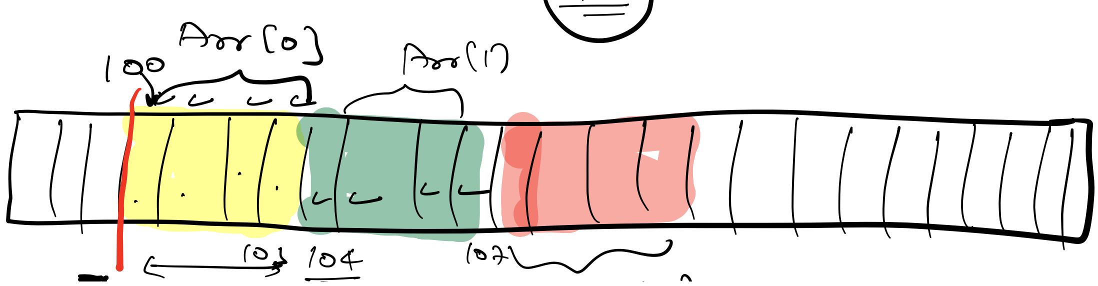
1 Byte = 8 bits 40

INT ⇒ 4 Bytes ⇒ 32 bits

array (integer) size 10

⇓

$4 \times 10 = 40$



Arr[1] Arr[2]

Arr ⇒ 0, 1, 2, 3, 4 → Arr[4]

Arr[0] Arr[2

Int arr[5] = {0, 1, 2, 3, 4} ;

100 ⇒ 104 ⇒ 108

Base address

i^{th} element
=0

i = 1	100	+	1×4
i = 2	100	+	2×4
i = 3	100	+	3×4
⋮			
i	100	+	i×4

Base + i× (size of one element)

N ⇒ N+1

O(1)

⇒ O(1) random access.

Basics

$$\begin{array}{l} \quad\quad\quad\quad\quad\quad\ \ 0 \quad 1 \quad 2 \quad 3 \quad 4 \end{array}$$

int A[ ] = {3, 2, 8, 9, 7}

1st element = 0
last Element = 4 } size = 5

Generalize $\underline{\underline{N}}$

1^{st} element : 0
Last Element : $n-1$

Accessing any index ⇒ O(1)

[0 , 1 , 2 , . . . n−1] (n)

Code

⇒ for (i = 0 ; i < n ; i++) { ⇒

print (Arr[i]) ; ⇒

}

$i \Rightarrow [0, n-1] \Rightarrow (n-1) - 0 + 1 = \underline{n}$

T.C. = $\underline{O(N)}$

Q Given an integer array of size N.

Count the number of elements having atleast 1 element greater than itself.

$$\begin{array}{cccccccccc} & & 0 & 1 & 2 & 3 & 4 & 5 & 6 & \\ A \Rightarrow & \{ & \boxed{3}, & -2, & 6, & 8, & 4, & 8, & 5 & \} \\ & & +1 & +1 & +1 & 0 & +1 & 0 & +1 & \end{array}$$

c = 0

max = ~~3~~ ~~6~~ 8

count = ~~0~~ ~~1~~ 2

Ans = 5

N = 7 − 2 = 5

Sol^n

① Brute force (Check everything)

```
count = 0;
for (i = 0; i < N; i++) {
    for (j = 0; j < N; j++) {
        if (A[j] > A[i]) {
            count++;
            break;
        }
    }
}
```

∀ element

Check if a greater element is present.

T.C. = $O(N^2)$

② Optimization

Observation :

Which element won't have any element greater than itself present in the array ??

⇒ Max Element.

⇒ Every other element will have atleast one element greater than itself (max element)

$$\text{Ans} = \underset{N}{\#\text{elements}} - 1$$

Total freq of max element.

Steps :

1) Find the max element
2) Count of max element (frequency)
3) Return answer.

Code

1)
```
int max = A[0];

for (i=1; i<N; i++) {
        if (A[i] > max) {
                max = A[i];
        }
}
```

T.C. = O(N)

2)
```
count_max = 0;
```

```
for (i = 0; i < N; i++) {        =>
    if (A[i] == max) {
        count_max ++;             =>
    }
}
```

$T.C. = O(N)$

$T.C. = O(N+N) = O(2N) = O(N)$

H.W. TODO :
How to do step 1 & 2
in one single loop.

Q.2 Given an array of size N.
Check if there exists a pair (i, j)
s.t. $A[i] + A[j] = \underline{K}$ ⇒ Given.

$i \,!= j$

$A[\,] = \{3, -2, 1, 4, 3, 6, 8\}$

indices: 0 1 2 3 4 5 6

4+4

$K = 10 \Rightarrow Arr[3] + Arr[5]$

$= 4 + 6 = 10$

$i = 3, \; j = 5$

$K = 8$ ✗

Soln ① Brute force (All possibilities)

⇒ for all pairs, find the sum and compare with K

$i \Rightarrow [0, N-1]$

$j \Rightarrow [0, N-1]$

$i = j$ not allowed

```
for (i=0; i<N; i++) { ⇒
    for (j=0; j<N; j++) { ⇒
        if (i != j) { ⇒
            if (A[i] + A[j] == K) {
                return (i, j);
            }
        }
    }
}
```

$T.C. = O(N^2)$

$i=2 \quad j=3$

$i=1, \quad j=2, 3$

$i=0 \quad j=1, 2, 3$

[0, N-2]

(i, j)

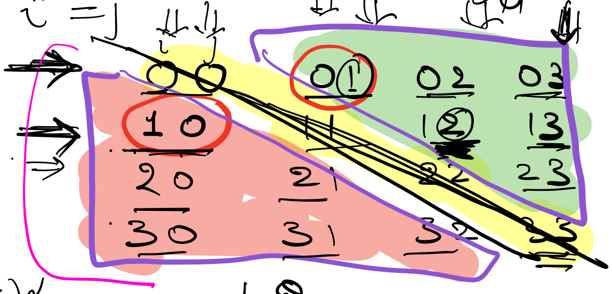
N = 4

i = j

$\Downarrow$

```
for (i=0; i<N; i++) {  ⇒
   for (j=0; j<N; j++) {  ⇒
      if (i != j) {  ⇒ ×
         if (A[i] + A[j] == K) {
            return (i,j);
         }
      }
   }
}
```

i = j



i = N-2

i = 0 , j = 1

i = 1 , j = 2

i = 2 , 3

i < N

i < N-1

```
for (i = 0; i < N; i++) {
⇒  for (j = i+1; j < N; j++) {
         if (A[i] + A[j] == K) {
            return (i,j);
         }
   }
}
```

i = N-1 ⇒ j = N ×

i = 3

i = 4

$i = 0 \qquad j = 1$

$i = 1 \qquad j = 2$

$i = 2 \qquad j = 3$

$i = 3 \qquad j = 4$

$\vdots$

$i = N-1 \qquad j = 0$

$[1, N-1]$

$[2, N-1]$

$[3, N-1]$

$j < 4$

$= N-1$

$N-2$

$N-3$

$N-4$

$\vdots$

1

0

$$T.C. = \frac{n}{2}\left(2a + (n-1)d\right)$$

$$AP = \frac{N}{2}\left(2(0) + (N-1)1\right)$$

$$= \frac{N}{2}(N-1) = \frac{(N)(N-1)}{2}$$

$$= \frac{N^2}{\cancel{2}} - \cancel{\frac{N}{2}}$$

$$= O(N^2) \Rightarrow$$

Q Given an array of size N.

Reverse the array

Const: No extra space allowed

$$\begin{array}{lcccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ A \Rightarrow & [-1, & 4, & 7, & 6, & -2, & 7, & 8, & 10] \\ R \Rightarrow & [10, & 8, & 7, & -2, & 6, & 7, & 4, & -1] \\ & 0 & 1 & & & & & N-2 & N-1 \end{array}$$

$$\begin{array}{ccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ A \Rightarrow & \{-1, & 4, & 7, & \cancel{6}, & \cancel{-2}, & 7, & 8, & 10\} \end{array}$$

$i = 0$

$j = N-1$

swap (A[i], A[j]);

i++;

j--;

$i = 0$	$j = 7$
$i = 1$	$j = 6$
$i = 2$	$j = 5$
$i = 3$	$j = 4$
$i = 4$	$j = 3$

$i > j \Rightarrow$ stop

$$\begin{array}{cccccc} & 0 & 1 & 2 & 3 & 4 \\ A = & [2, & 6, & 3, & 4, & 5] \end{array}$$

$i = 0$	$j = 4$
$i = 1$	$j = 3$
$= i = 2$	$j = 2$

$\Rightarrow$ No swap.

$i >= j \Rightarrow$ stop

Code

$i = 0;$

$j = N-1;$

$\Rightarrow$ while (i < j) {

H.W. swap (A[i], A[j]) ; $\Rightarrow$ (O(1))

i++;

j--;

}

T.C. =

$i \Rightarrow [0, N/2] \Rightarrow N/\cancel{2}$

T.C. = O(N)

if (N % 2 == 0)

(1 <= n)

sqrt(n) (n+1) (n+1)

$(\sqrt{n})(n^2) \times c$

$i = 0$

while ($i \times i < N$) { $\sqrt{N}$ times)

for (j = 0 ; j < N ; j++) {

for (k = 0 ; k < N ; k++) {

}

}

$O(N^2)$

i++ ;

}

$i \times i < N$

$i^2 < N$

$i < \sqrt{N}$

$i = [0, \sqrt{N} - 1] = (\sqrt{N} - 1) - 0 + 1$

$= \sqrt{N}$

TC 2 H.W.

$i = 0$

```
while ( i*i <= N ) {
    for ( j = 0 ;  j <= N ;  j++ ) {
        for ( k = 0 ;  k <= N ;  k++ ) {
            O(1)
        }
    }
}
```

$O(N^2)$

N^2

$i = 0 \qquad j \Rightarrow [1, N-1] \Rightarrow N-1$

$i = 1 \qquad j \Rightarrow [2, N-1] \Rightarrow N-2$

$i = 2 \qquad j \Rightarrow [3, N-1] \Rightarrow N-3$

$\vdots$

$i = N-1 \qquad j \Rightarrow \times \qquad 0$

$\cancel{0} + 1 + 2 + 3 \; \ldots \; N-1$

$$\frac{(N-1)(N-1+1)}{2} = \frac{(N)(N-1)}{2}$$

$A = 1$

$N = N-1$

$D = 1$

$$S_n = \frac{(N-1)}{2}\left(2 + (N-1-1)\,1\right)$$

$$\left(\frac{N-1}{2}\right)\left(\cancel{2} + N - \cancel{2}\right)$$

$$= \frac{(N)(N-1)}{2} = O(N^2)$$

limit $\Rightarrow$ Not req

size $\Rightarrow$ INT $\Leftarrow$ max value of INT

$2^{32}-1$ $\Leftarrow$

```
⇒ for (i=N; i>0; i=i/2) {
     → for (j=0; j<i; j++) {

       }
  }
```

$N \Rightarrow [0, N] \Rightarrow N$

$N/2 \quad [0, N/2] \Rightarrow N/2$

$$N/4 \Rightarrow [0, N/4] \Rightarrow \begin{array}{c} N/4 \\ N/8 \\ N/16 \\ \vdots \\ 1 \end{array}$$