Sorting

Content

- -> Understanding Sorting
- Jew problem en Sorting
- -> 1 sorting algo
- -> Comparator function

Sorting ?

Arranging data in increasing / decreasing order based

on some parameter.

of 2 4 7 11 15: sorted in Asc, persone kr = angly volves

: sorted in Dese, par = array values

1237 496: sorted in Ase based on # factors.

3 3 4 par = # of factor of array values Afactors 1 2 2 2

Inbuilt library language sorter is present. How()? logic? > { In Advanced Bortel } n = no. of elements to sort TC: O(nlogn) Gustien 1: Elements Removed Cinen N elemens, at every step remove an array clement. Cost of remove element: som of all elements present in array find min. west to remove all elements.

first calculate fere cost, then remove the element.

1 4 3 9 9(3) = { 2

Cost

remove 2: 2+1+4=7

1+4=5 remove 1: 4

remove 4:

7+5+4=16 total cost =

remove 1: 7

remove 2:

remove 4:

total wst = 17

```
remove 4: 7
remove 2: 3
remove 1: 1
 total Wy = 11
```

remove 6: 15

remove 9: 9

remove 3:5

remove 2: 2

total cost = 31

Observation: deleting element by element in decreasing order gives the min. cost?

a(4) = { a, b, e, d }

Temore α : $\alpha + b + c + d$ Temore b: b + c + dThe cost of the cost of remove d: d

a+20+3c+4d

```
ind min Cost (all) }
    n=a.leugth
    Sort (a, DESC); - sort an in desc order
                      La TODO in your own language
    aus=0
   tor (i=0; i<n; ++i) }
                              TC:OKN)
       aus += au) x(i+1)
                             TC: OCHIOSN+H)
                               : O(NIOgN)
   return am
                             SC: 0(1)
     0 1 2 3 9 6 2 4
      Surt(9) = 6 4 3 2
      am = 6x1 + 4x2 + 3x3 + 2x4
           = 6 +8+9+8
           = 31
```

Suestion 2: Noble Integers & Flements are unique 3 liver N elements, calculate no. of noble integers. An element in all is said to be noble if Sno. of elements Lele = ele itself 3 eg ? -1 +5 3 5 +10 43 #1cm 2 1 3 5 0 4 Aus=3 $\frac{1}{2}$ $\frac{1}{3}$ $\frac{1}{2}$ $\frac{1}{3}$ Ans=1 #1cm 0

Brukbru:

for every element, iterate & get # ele < ele and compare with ele. iteelf.

int auszo

forli=0; i<n; ++i) } TC: OCN2) count=0 1 #ex < ali) SC:0(1) forlj=0; j~n; ++j) 3 if (alj) <ali)

```
if (all) = = count')
             ++aW
Idea: Sort the array in asc. order
 Sorted (a): a10) a11) a12) ... - a li-1) ali) ali+1) ....
               Lau these no. are
less than a(i)
                                 =) if (au) ==i)
                                         fuen
aui) is noble
  int mobile ( al)) &
       n=a.rength
       Sort (a, ASC) -> 70DU
                                       TC: O(NIGN)
       ans =0
                                       S(:0(1)
        for (ia); i<m', ++i) }
           if (all) == i)
        return am
  dry rm: -1 -5 3 5 -10 4
                                            Ans = 3
```

Question 3:

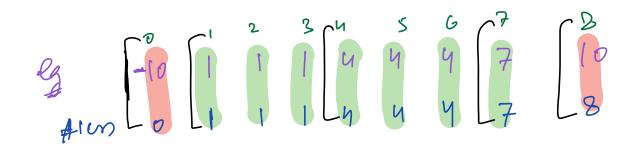
Count Noble elements: § Elements can repeat 3 Duplicate values

We can solve veing bruteforce: TC: O(N2) S(:O(1)

9 0 2 2 3 3 6 Hen 0 1 1 3 3 5

Aus=3

Above Indexing approach will not work



Observation 1: index = #1em is correct for only the
first occurance of every element

Observation 2: if on element is noble, all occurrances are noble.

```
Ideq: if element comes for the first time

Lound of ele. less than ali) = i

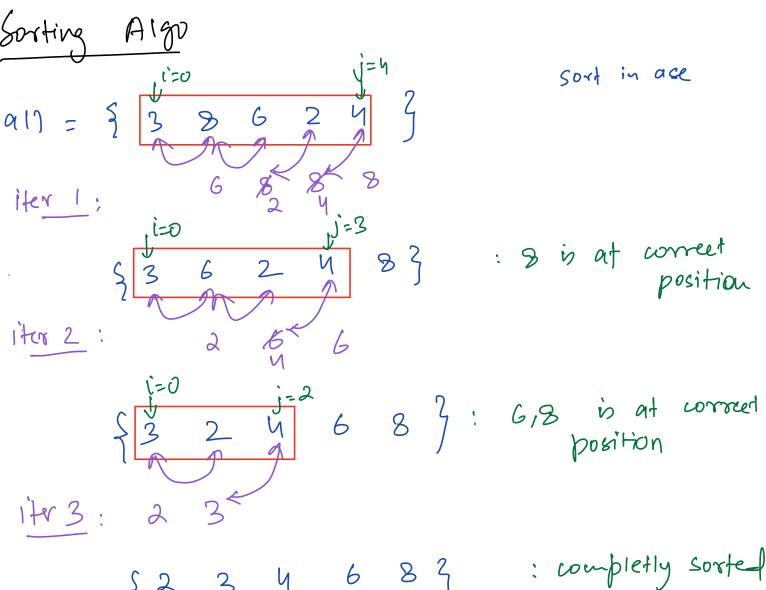
if element repeals -> if (ali) == ali-1)

count of ele. less than ali) will be

barn on prev. one.
```

int noble (all) } n= a. length aus=0 Sort (a, ASC); - TODO if (alo) = =0) & ans = 1 } TC: O(NIOgN) len =0 / #ere < arij SC:001) for (i=1; i<n; ++i) { if (ali)!=ali-1]) { // aci) coming for first len =i if (ali) == 1en)

return aus



Sort (all)
$$\S$$
 $n = q \cdot |ergth$
 $for(j = m-1; j \neq 0; --j) \S$
 $for(i = 0; i \leq j; ++i) \S$
 $if(ali) > ali+1])$
 $Sump(ali), ali+1])$
 $\S(0, 0)$
 $\S(0, 0)$

for any type of sorting like soot based on # factors, we use comparator function.

for (j=n-1; j>0; --j) }

for (i=0; i<j; ++i) }

if (compare (ali), ali+1]) to get your swep(ali), ali+1])

swap(ali), ali+1])

desired sorting.

```
In Java / JS
Comparator workforn Comparator = ( Integer a, Integer b) >> {
   / If you want a to come before b: return -1
   I If you want all are same: return o
   / If you want b to come before a: return )
 Sort (a, custom comparator);
 Say we have to sort on acc. order of # of factors
Comparator custon Confearation (Integer a, Integer b) -> §
      fa > factors (a) -> O(JN)
       fb = factors (b)
       if (fa < fb) return -1
                           TC to sort:
         return 1
                             O( N103N x IN)
```

```
In Python
 def custon Comparator (a, b):
    / If you want a to come before b: return -1
     I If you want all are same: return o
    / If you want b to come before a: return )
  a. sort ( key = cmp- to- Key (custom Comparator))
In C/C++
  bool wstomConferator (int a, int b) }
    I If you want a before b: return true
                                · return false
     11 E18C
 Sort (n, custon Comparator)
```