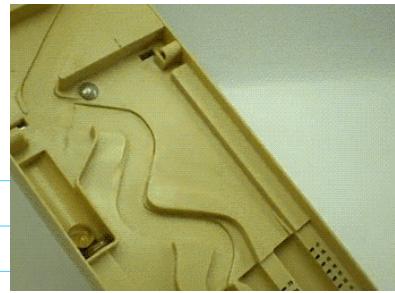


Topics 8-sorting intro  
 -min/avg remove  
 -new element  
 -custom cmp



## Sorting

arrangement of data

in a specific order on the basis of same parameter

ex  $a[] = \{ -1, 0, 3, 7, 9, 27 \}$  ans = yes ✓ sorted

ex  $b[] = \{ 100, 56, 23, 12, 9, 0, -10, 45 \}$  ans = yes : sorted ✓

Quiz  $c[] = \{ 1, 13, 9, 6, 12 \}$   
 #of factors 1 2 3 4 6

Why sorting? → search easier, present data better  
 optimize algorithm, analyze

How to sort? sort Algorithm advance class 13-15

Java

`int[] a = {1, -1, 5, ..., 7};`

`Arrays.sort(a);`

Python

`a = [1, 2, 5, -3]`

built-in

`a.sort()`

some characteristics of built-in sort algorithms:

ebay

a) criteria can be an input. same

Sort algo,  
 different  
 criteria

\$ →

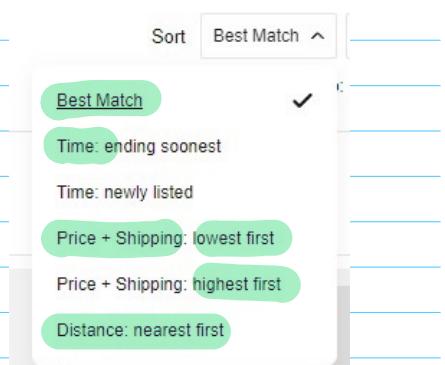


diameter →



alphabetical →

Dime, Nickel, Penny, Quarter



b) Most of sort Algos. are comparison based  
In particular they compare two data

<https://www.toptal.com/developers/sorting-algorithms>

so what?

building  
of  
customizing  
a sort  
algo.

c) TC  $\geq \Theta(n \log n)$  SC  $\geq \Theta(n)$

$\Theta(1)$

$\Theta(n)$

int ← output ← min cost

P1 Given an array of integers . you have to delete all the elements from the array! <sup>how?!</sup> minimize cost

Removing each element has a cost : sum of all elements left in the array (including the item being removed)

ex1 $a: \{2, 1, 4\}$ Case 1 <small>possible order not min cost</small> <del>2 + 4</del>	Case 2: <small>best answer</small> $\begin{array}{l} ① \text{delete } 4: 4+1+2=7 \\ ② \text{delete } 2: 2+1=3 \\ ③ \text{delete } 1: 1=1 \end{array}$
$\begin{array}{l} ① \text{delete } 1: 1+2+4=7 \\ ② \text{delete } 2: 2+4=6 \\ ③ \text{delete } 4: 4=4 \end{array}$	<small>ans</small> $\begin{array}{l} ① \text{delete } 4: 4+1+2=7 \\ ② \text{delete } 2: 2+1=3 \\ ③ \text{delete } 1: 1=1 \end{array}$

Quiz

$$\{4, 6, 1\}$$

6	6 + 4 + 1 = 11	17 ← ans
4	4 + 1 = 5	
1	1 = 1	

Quiz

$$\{3, 5, 1, -3\}$$

5	5 + 3 + 1 - 3 = 6	2 ← ans
3	3 + 1 - 3 = 1	
1	1 - 3 = -2	
-3	-3 = -3	

ex  $\{a > b > c > d\}$

assume sorted  
desc

$$\begin{aligned} \times a &: a+b+c+d \\ \times b &: b+c+d \\ \times c &: c+d \\ \times d &: d \end{aligned}$$

→ descending order  
 $a+2b+3c+4d$   
 $a > b > c > d$

Quiz

```
int minCost( int a[]){
```

TC: O(n log n)

SC: O(n)

$n = a.length$

cost = 0

Arrays.sort(a, descending)  $\mathcal{O}(n \log n)$

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

$n | cost += (i+1) \times a[i]$

}

ret cost

}

TC<sub>8</sub>

TC<sub>8</sub>

$O(n)$

$n \log n + n$

P2.1 Find Count of noble integers in a given array:

Noble ints: count of elements smaller than  $A[i] \geq A[i]$

ex { $\overset{0}{1}, \overset{1}{-5}, \overset{2}{3}, \overset{3}{5}, \overset{4}{-10}, \overset{5}{4}$ } No duplicate  
all  $A[i]$  unique

Count #smaller 2, 1, 3, 5, 0, 4 ans=3

Quiz { $\overset{0}{-3}, \overset{1}{0}, \overset{2}{2}, \overset{3}{5}$ } ans=1  
Count #smaller 0 1 2 3 J

not optimized

Quiz int nobel(count1( int a[])){  
 TC: O( $n^2$ ) SC: O(1)  
 ans = 0; n = a.Len  
 for (i=0; i < n; i++) {  
 countSmaller = 0  
 for (j=0; j < n; j++) {  
 if (a[j] < a[i]) {  
 countSmaller++  
 }
 }
 if (countSmaller == a[i]) ans++ // found noble
 }
 ret ans
}

if array is sorted a/cs. # of elements smaller than  $a[i]$  is  $i$

Optimization ideas

Sorted  $a[0] = a_0$   
 $a_0 < a_1 < a_2 < \dots < a_{i-1} < a_i < a_{i+1} < \dots < a_{n-1}$   
 $[0 \dots i-1] a_i [i+1 \dots n-1]$   $a[i] \geq ? = i$

**Quiz**      `int nobelCount2(int a[]){`

TC:  $O(n \log n)$       SC:  $O(1)$

```

n = a.length
Arrays.sort(a) //  $n \log n$       SC:  $O(1)$ 
ans = 0 // no of Nobels
for(i=0; i < n; i++){
    if(a[i] == i) ans++
}
ret ans
}

```

$n \log n + n$

$[0, 1, 2, 3, 4]$   
 $0 \quad 1 \quad 2 \quad 3 \quad 4$

P2.2 \* what if we have duplicates?

**Quiz**       $\{-10, 1, 1, 3, 100\}$

Count #smaller       $\begin{matrix} 0 & 1 & 1 & 3 & 4 \\ 0 & 1 & 1 & 3 & 4 \end{matrix}$        $ans = 3$

**Quiz**       $\{-10, 1, 1, 2, 4, 4, 4, 8, 10\}$        $ans = 5$

Count #smaller       $\begin{matrix} 0 & 1 & 1 & 3 & 4 & 4 & 4 & 7 & 8 \\ 0 & 1 & 1 & 3 & 4 & 4 & 4 & 7 & 8 \end{matrix}$

**① if ( $a[i] \neq a[i-1]$ ) use  $a[i] \geq i$  for nobel**

**sorted**       $\begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\ -3, 0, 2, 2, 5, 5, 5, 5, 8, 8, 10, 10, 10, 14 \\ 0 \quad 1 \quad 2 \quad 2 \quad 4 \quad 4 \quad 4 \quad 4 \quad 8 \quad 8 \quad 10 \quad 10 \quad 10 \quad 13 \end{matrix}$        $ans = 7$

Count #smaller  
 $\downarrow$   
 check and  
 increase  
 time

if array is sorted aks. # of elements  
 smaller than  $a[i]$  is  $i$

Quiz

TC:  $O(n \log n)$

SC:  $O(1)$

```
int nobel(count3( int a[] )) {  
    n = a.Len  
    Arrays. Sort(a) //  $O(n \log n)$   $O(1)$   
    ans = 0 // no. of nobels  
    countSmaller = 0  
    if( a[0] == 0 ) ans++  
    for( i=1 ; i < n ; i++ ) {  
        if( a[i] != a[i-1] ) {  
            countSmaller = i  
        }  
        if( a[i] == countSmaller ) { // nobel?  
            ans++  
        }  
    }  
    ret ans  
}
```

$n \log n + n$

P3 Sort a given array of integers based on count of factors of each int. If count of factors are equal, sort base on the magnitude of the int.

ex { 9, 3, 10, 6, 4 }  
 3 2 4 4 3

3, 4, 9, 6, 10  
 2 3 3 4 4      ans ↵

int[] a = new int[]{ 9, 3, 10, 6, 4 }

Arrays.Sort(a, cmp)      what is this? haven't seen this before!

Custom Comparator (we can also override the Comparer for the object in java & C#)

Java, Python, JS, C#, ...

what is this?  
 what is the diff.  
 with simple "cmp"

should ← int cmp(T obj1, T obj2){

I put obj1 if obj1 after obj2 ret +ve (e.g. +1, +2, +99, etc.)

before obj2 if obj1 before obj2 ret -ve (e.g. -1, -2, -99, etc.)

if obj1 == obj2 ret 0

e1 e2 e3

① any +ve or -ve is acceptable? or just -1, +1

② what if we don't cover all cases?

③ how does the sort func. know only based on "cmpn"?

C++ bool cmp(int a, int b){

① in sorted format if a comes before b, return true

② otherwise false.

should

I put a

before b

}

1, 3, 5

5, 3, 1, 1

int

ex descending Order

before after C++

should

a come  
first?

bool cmp(int a, int b){

ret a>b

}

5 3

☒

3 1

int cmp(int a, int b){  
    if(a>b) ret -1  
    if(a<b) ret +1  
    if(a==b) ret 0  
        true  
        false

ex array of strings, sort based on their len, ascending

bool cmp(string s1, string s2){

ret s1.len - s2.len

}

s1                    s2  
"aa"                "abc"

,"abcd"

4

{ if(s1.len < s2.len) ret -1  
    if(s1.len > s2.len) ret +1  
    if(s1.len == s2.len) ret 0  
        -1, 0, +1  
        a, b

bool cmp(int a, int b){ //for count of factors

af = countOfFactors(a)

bf = countOfFactors(b)

if(af < bf) ret -1

if(af > bf) ret +1

if(af == bf){

    if(a < b) ret -1

    if(a > b) ret +1

    if(a == b) ret 0

}

}