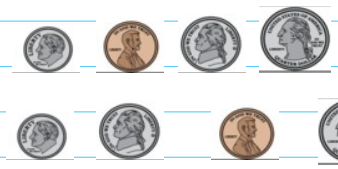
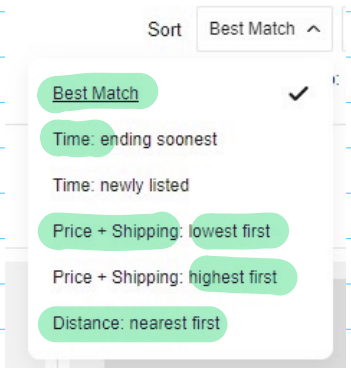
- Intro to cmp
- factor sorting
- largest number
- B closest point

criteria can be an input. same sort algo, different criteria



Dime, Nickel, Penney, Quarter

e-commerce search result



- Sort an object:

```
Class product{
    productID
    name
    price
    origin
    time
     :
     :
}
```

Comparators.

- what are they? where do we need them?
- C++ style
- Java, Python, JS, C#, ...

- Sort but not necessarily ascending:

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

Arrays.Sort(a) → a = {3,4,6,9,10}

→ ascending

Sort Algorithm: I can sort however you want. you just tell me: between two objects which one came first. → how

↳ recap on merge sort & quick sort comparision

C++ cmp 的 style

```
int[] a = new int[]{9,3,10,6,4}
Arrays.sort(a, cmp)

bool cmp(int a, int b){   ← Should a come befor b ?
    if a comes before b return true
    otherwise return false.
}
```

10, 9, 6, 4, 3

descending : larger element comes first before smaller element

```
bool cmp(int a, int b){   Should a come befor b ?
    if(a>b) ret true
    else ret false
}
```

== ret a>b

Larger item can
① first
② 2nd

↳ a<=b

ret a<b → ascending

P1 array of strings, sort based on their len, ascending

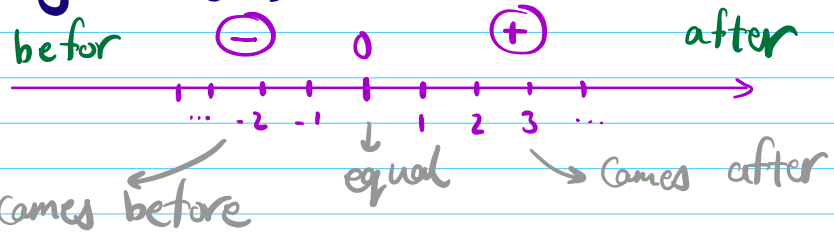
{"abc", "zz", "xx", "aXYb"} → "xx", "zz", "abc", "aXYb"
(3, 2, 2, 4)

unstable

```
bool cmp(string a, string b){   // should a come befor b?
    aL = a.Len; bL = b.Len
    if(aL<=bL) ret true
    else ret false
}
```

ret a.Len < b.Len
<=

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



10,9,6,4,3

① int descending

```
int cmp(int a, int b){   // a comes
    if(a>b) ret -1       // -10
    if(a==b) ret 0
    if(a<b) ret +1       // +909
}
```

10,9,6,4,3,3

before −
eq (doesn't matter) 0
after +
b

ret b-a → desc
ret a-b → asc

Quiz

{"abc", "zz", "xx", "aXYb"} → "xx", "zz", "abc", "aXYb"
(3, 2, 2, 4)

② str len ascending

```
int cmp(string a, string b){   // a comes
    aL = a.Len; bL = b.Len
    if(aL<bL) ret -1
    if(aL==bL) ret 0
    if(aL>bL) ret +1
}
```

before −
eq (doesn't matter) 0
after +
b

ret a.Len - b.Len

P2 Sort a given array of integers based on count of factors of each int. if Count of factors are equal, sort base on the magnititud of the int. ascending

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

3 4 9 6 10
2 3 3 4 4

Quiz
TC: $(n \times \log(n) \times \sqrt{\max(a[i])})$
SC: → O(n)

```
a.sort(a, cmp)

int cmp(int a, int b){ // for Count of factors
    aF = CountOfFactors(a)   → O(√max(a[i]) x 2
    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 0      } ret a-b
        if(a>b) ret +1
    }
}
```

---

lexicographical order : Dictionry order

"aa"
"aap"
"abc"
"abcd"
"abz"
"bcd"
"z"
"za"

add
adding ↓ larger

"3" ← smaller
"30" ←
"34" larger ↑

$0 <= a[i] <= 2*10^9$

P3 Given an array a of integers, arrange them such that they form the largest number. output → string representation of the largest number

ex {2, 3, 9, 0} 9320 ans

ex {3, 30, 34, 5, 9} 9534330

9534303 wrong ✗

① 0 z a z

② 0 z z a

ex {0, 0, 0} → "0"

000 → "000" ← HW

303 330

3 30

330 303

9 34

349 934

9999 9998

Quiz

TC:

SC:

```
string LargestNum(int a[]){
    Array.Sort(a, cmp)
    ans = a[0].toString()
    for(i=1; i<a.Len; i++){
        ans += a[i].toString()
    }
    ret ans
}
```

x has bug

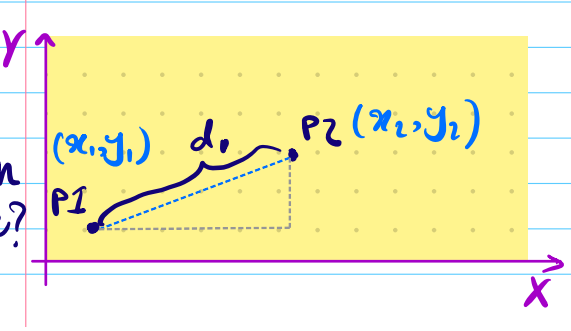
```
int cmp(a, b){
    ret -String.Compare(
        a.toString()+b.toString(),
        b.toString()+a.toString()
    )
```

$n \log n \times \underbrace{\log_{10}(\max(a_i))}_{\text{len of int to str}} \times 2$

d_1 d_2

$d_1 = \sqrt{16}$ $d_2 = \sqrt{17}$ 16 17

Euclidean distance?



$d = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$

↳ sqrt(...)

* Answer will be unique

P4 Given a list of points (x,y) on 2D plane, find the B closest point to the origin.

B >= 1

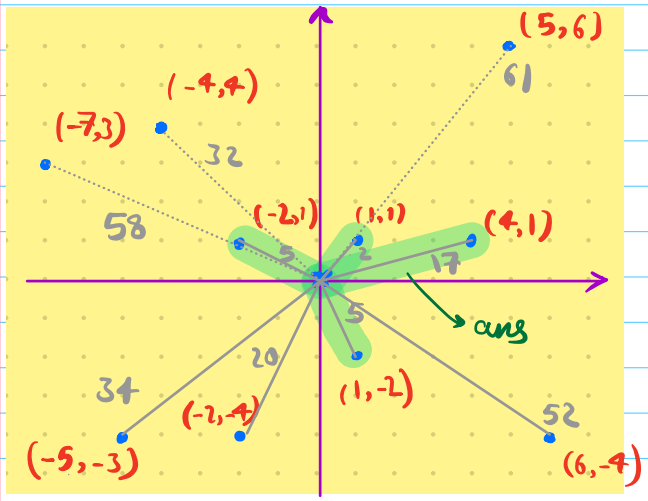
ex

d^2

B = 4

what if all are on circle and more than 4 points

* Answer will be unique



input format

A

	x (0)	y (1)
0	-10	3
A[1] → 1	7	-30
2	-100	-35
⋮	+99	+72
⋮	⋮	⋮
n-1	0	93

x = A[1][0]

y = A[1][2]

{58, 32, 5, 2, 17, 61, 34, 20, 5, 52} → {2, 5, 5, 17}

Bx2

```
int[][] bClosest(int[][] A, int B){
    Arrays.sort(A2, cmp)        A2 → Convert → list<Pair<x,y>>
    ans = new int[B][2]
    //Copy the first B row from the sorted arr
}
```

Quiz

TC: nlogn

SC: depends on sort

```
int cmp(Pair a, Pair b){
    ret (a.x^2 + a.y^2) - (b.x^2 + b.y^2)        → O(1)
}
```

distance from (0,0)

Why (-) → a - b default asc sort

$\log_{10} 999 = 2.9$ (8)

$\log_{10}(8^{1.9})$ (7)

$\log_{10} 9^{0.8} = 0.9$

$\log_{10} 1000 = 3$

$\log_{10} 100 = 2$

$\log_{10} 10 = 1$

no of digits of int $n \rightarrow \lceil \log_{10} n \rceil$