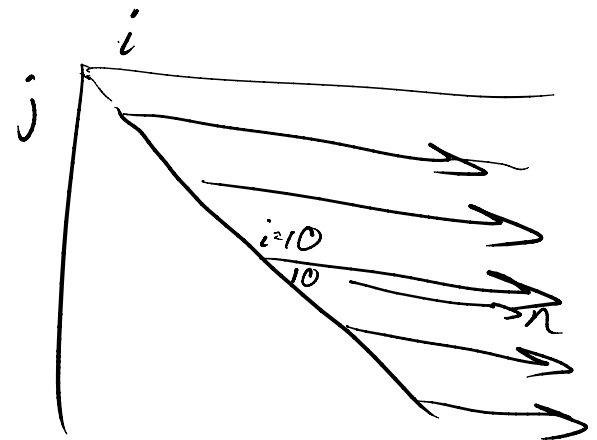
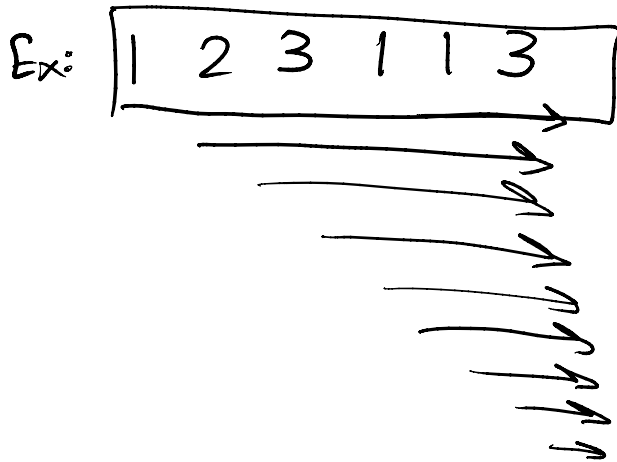


input: array of ints.

challenge: count "good" pairs i, j

$$\text{arr}[i] == \text{arr}[j] \& i < j$$

easiest solution	2-D forloop
<pre> for i n for j = i+1 ... n if arr[j] == arr[i] count++ comparing i and j </pre>	



Dictionary

Key $\rightarrow$ value

dict[10]
["h"]

new dict()

dict[100000000] = 10

size 1

Array

Index $\rightarrow$ value

arr[10]

new arr[]

arr[] = 10

size huge

arr = ^{0 1 2 3 4 5} 1 2 3 1 1 3

dict == Hashmap in Java

dict = new dict()

for (i = 0 i < n i [↓]++) {

if (dict.containsKey(arr[i])) {

// this means we have seen arr[i] before

count = count + dict.get(arr[i])

}

else { // this is the first arr[i] we are encountering

dict.put(arr[i], 0)

}

dict.put(arr[i], dict.get(arr[i]) + 1);

}

// increase # of arr[i] we have seen.

Example input but with letters = A B C A A C

i = 0

A

count = 0

dict {

A: 1

← after i=0 we've seen 1 A

}

i = 1

A B

count = 0

dict {

A: 1

B: 1

← after i=1 we've seen 1 A & 1 B

}

$i=2$ ABC
dict {
A:1
B:1
C:1
}

count=0

$i=3$ ABCA

count=1 ← we've seen 1 A according to dict,
so add 1 good pair to the count

dict {
A:2
B:1
C:1
}

$i=4$ ABCAA

count=1+2 ← we've seen 2 A's according to dict,
add 2 to the count

dict {
A:3
B:1
C:1
}

$i=5$ ABCAAC

count=1+2+1 ← add 1 for the 1 C that we've seen

dict {
A:3
B:1
C:1
}

{ C:2

total count = 4

side effect: dict shows total # of each letter
in the input, 3 A's 1 B and 2 C's