Wednesday Oct. 17

Lecture 11

- Lab Test 2 : October 29

  Study Guide availabe next Monday

$O(100n)$ vs. $O(2n)$

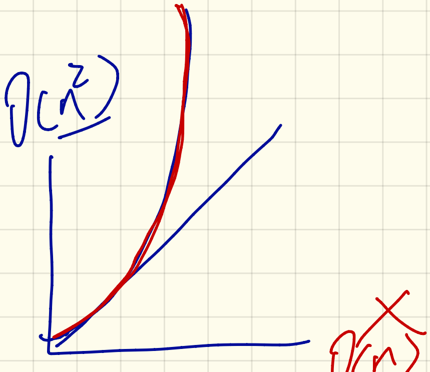
$O(n^0) \subset O(n^1) \subset O(n^2) \subset \cdots$

$O(2^n)$

$2n$ is $O(n)$ is $O(n^2)$ is $O(n^3)$

$2n + 100 \cdot \log n$

$2n$ is $O(n^2)$

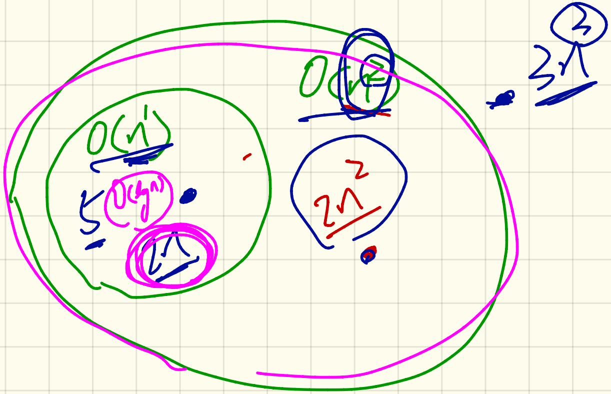


$O(n)$ $O(n^2)$ is $\cancel{O(n)}$

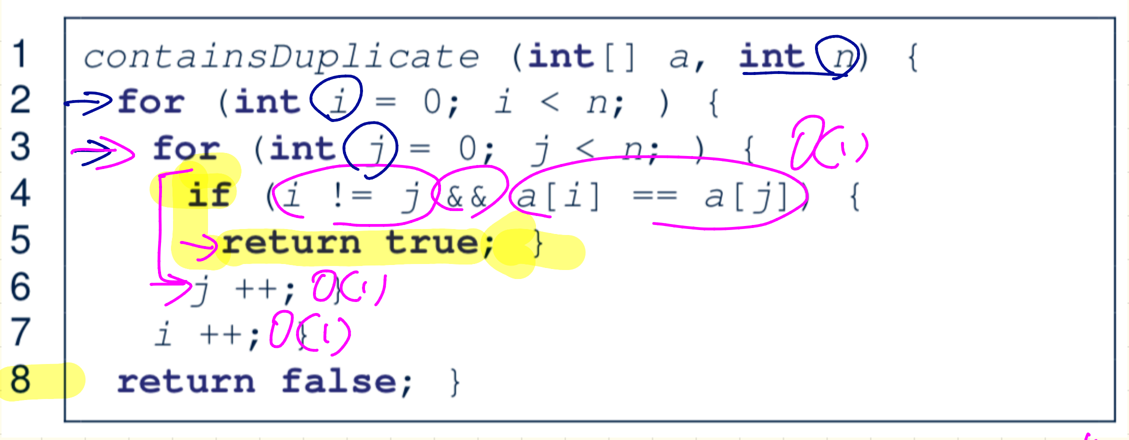
$c = 2 + 100 = 102$

n_0	$2n + 100 \cdot \log n$	$102 \cdot n$
1	$2 + 0 = 2$	≤ 102

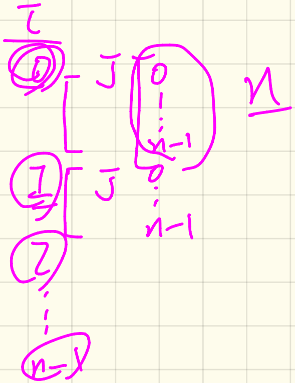
$O(n^1)$ vs. $O(n^2)$



Determining Asymptotic Upper Bound (1)

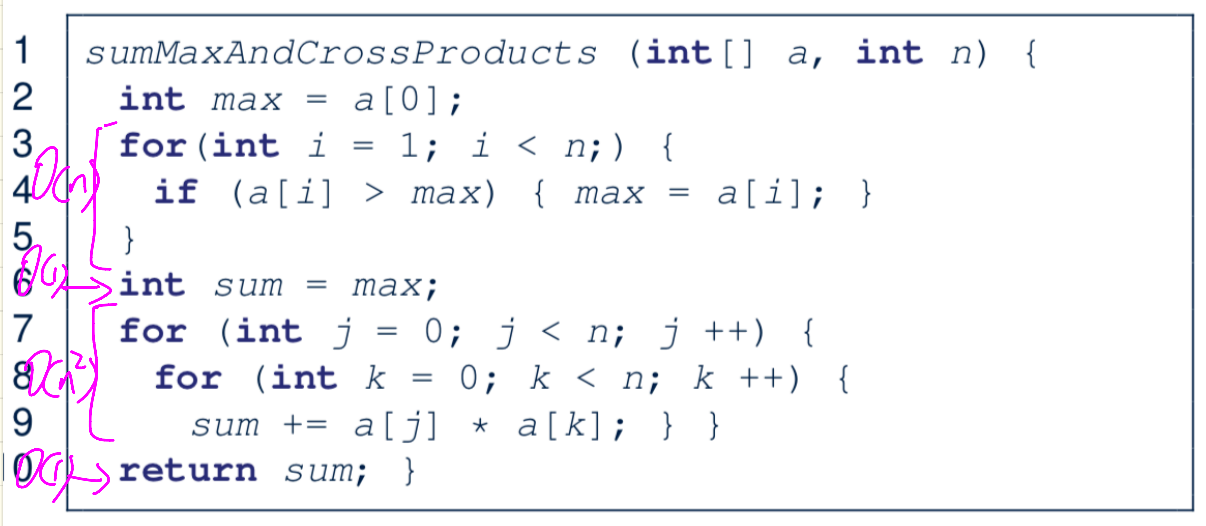


```
containsDuplicate (int[] a, int n) {
  for (int i = 0; i < n; ) {
    for (int j = 0; j < n; ) {  O(1)
      if (i != j && a[i] == a[j]) {
        return true; }
      j ++; O(1) }
    i ++; O(1) }
  return false; }
```



$$O(\underbrace{1}_{\text{body of loop}} \times \underbrace{n}_{\text{possible values of j for each i}} * \underbrace{n}_{\text{possible values for i}}) = O(n^2)$$

Determining Asymptotic Upper Bound (2)



```
sumMaxAndCrossProducts (int[] a, int n) {
  int max = a[0];
  for(int i = 1; i < n;) {
    if (a[i] > max) { max = a[i]; }
  }
  int sum = max;
  for (int j = 0; j < n; j ++) {
    for (int k = 0; k < n; k ++) {
      sum += a[j] * a[k]; } }
  return sum; }
```

$O(n)$

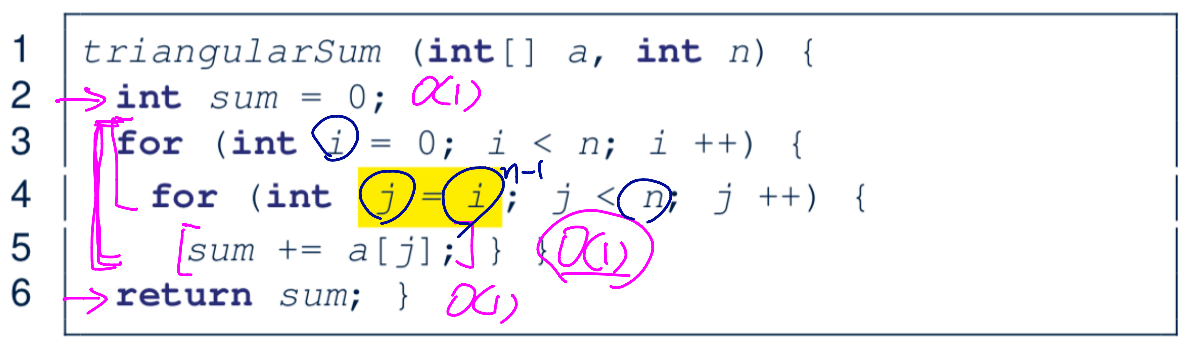
$O(1)$

$O(n^2)$

$O(1)$

$$O(n \underline{+} n^2) = O(n^2)$$

Determining Asymptotic Upper Bound (3)



```
triangularSum (int[] a, int n) {
  int sum = 0;  O(1)
  for (int i = 0; i < n; i ++) {
    for (int j = i; j < n; j ++) {  n-1
      sum += a[j]; } }  O(1)
  return sum; }  O(1)
```

I

0 [J 0 ⋮ n-1] } n

1 [J I ⋮ n-1] } $n-1$

2 ⋮

⋮ ⋮

n-1 [J=n-1] } 1

$$n + (n-1) + \cdots + (1) = O(n^2)$$

```
m( int[] a , int n) {
    for( ; ; ) {
    }
}
```

$i=0$ $i<n^2$ $i++$

O(1)

O(n^2)

O(n) ?

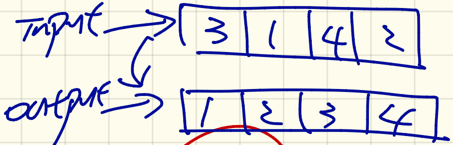
X

2	3	1	4	6

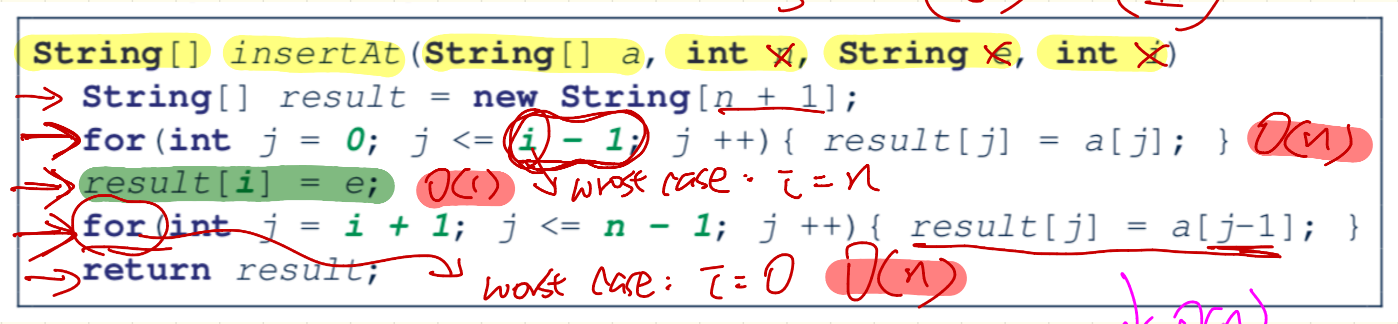
7

2	3	1	4	6	7

Inserting into an array



3 "Jim" 1



```
String[] insertAt(String[] a, int n, String e, int i)
  String[] result = new String[n + 1];
  for(int j = 0; j <= i - 1; j ++){ result[j] = a[j]; }
  result[i] = e;
  for(int j = i + 1; j <= n - 1; j ++){ result[j] = a[j-1]; }
  return result;
```

O(n)

O(1)

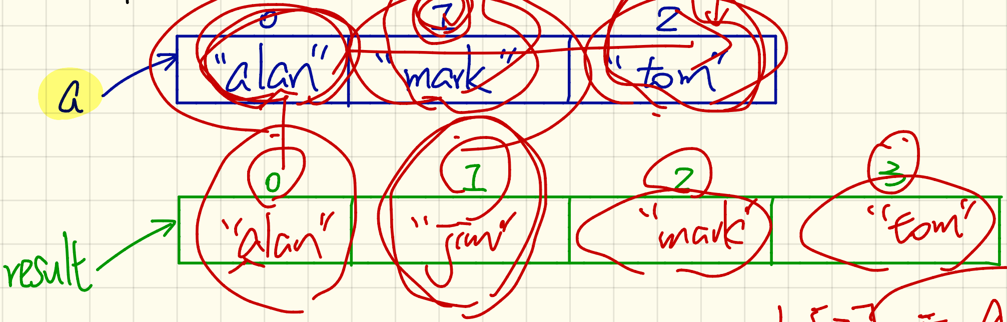
wrost case : i = n

worst case : i = 0

O(n)

O(n)

Example : insertAt({"alan", "mark", "tom"}, 3, "jim", 1)



RT ?

result[2] = a[1]

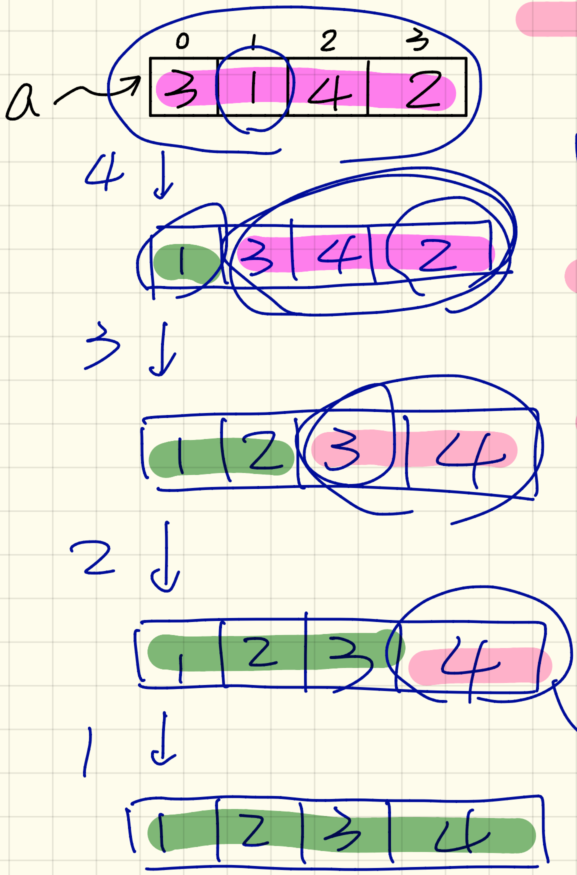
[3] = a[2]

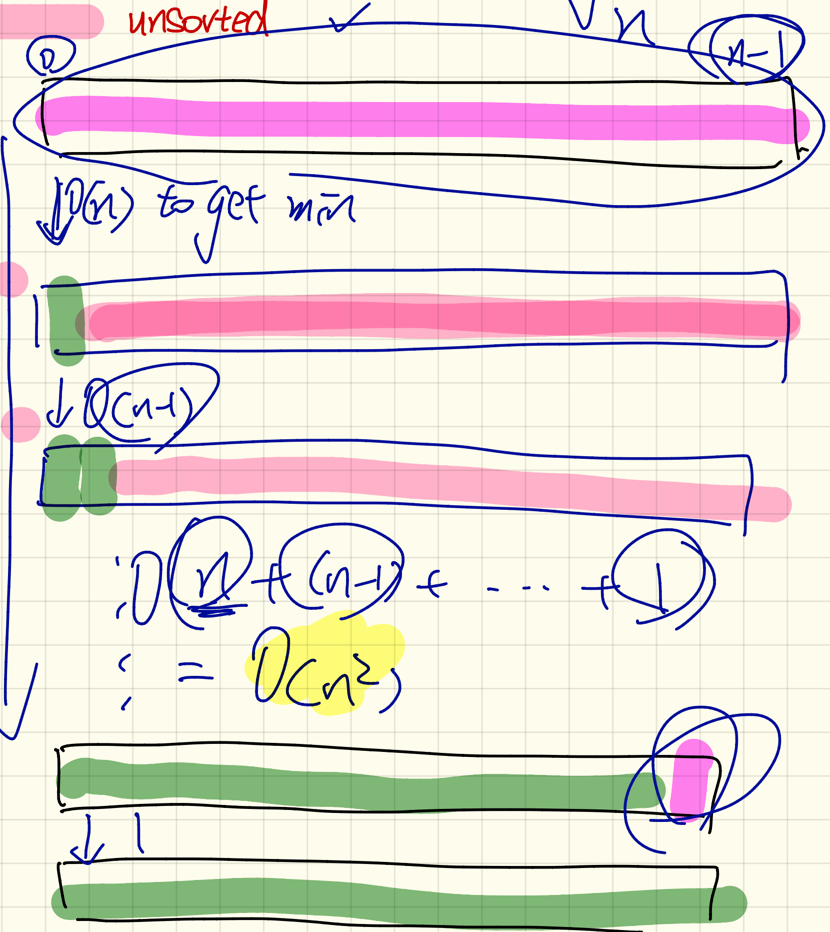
Selection Sort : Idea

🟩 Sorted ✓

🟥 unsorted ✓

How many selections?





O(n) to get min

O(n-1)

$O(n + (n-1) + \cdots + 1)$

$= O(n^2)$

Insertion Sort : Idea

🟩 sorted

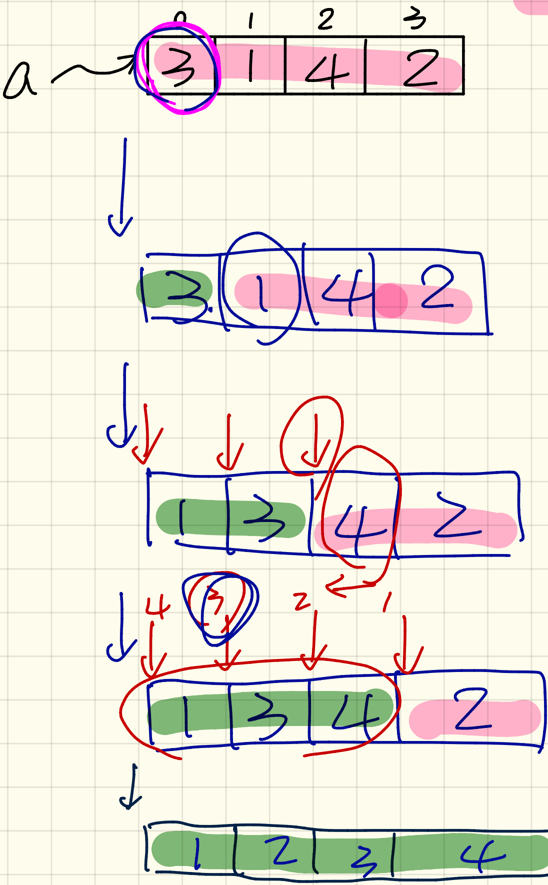
🟥 unsorted

$$O(1+2+\cdots+(n-1)) = O(n^2) \quad \boxed{1000} \quad \underline{1M}$$

pick the left-most element of 🟥

insert it to the correct spot

in 🟩



a
0
1
2
3
3 1 4 2
3 1 4 2
1 3 4 2
4
3
2
1
1 3 4 2
1 2 3 4