Friday, February 3

Written Test 1 Review

* k<n only executed before outer loop exits (when j=n)

Count # of Primitive Operations

```
int sumMaxAndCrossProducts (int[] a, int n) {
  int max = a[0];                                   2
  for(int i = 1; i < n; i ++) {                     i=i+1
    if (a[i] > max) { max = a[i]; }
  }
  int sum = max;                                    1
  for (int j = 0; j < n; j ++) {                    j=j+1 (2)
    for (int k = 0; k < n; k ++) {                  (2)
      sum += a[j] * a[k]; }                         (5)
                                                    sum = sum + a[j] * a[k]
  return sum; }                                     1
```

$i=1$	$i<n$
1	T
⋮	F
$n-1$	
n	

1. # times $i<n$ evaluated? (n)
2. # times body of loop exec? $(n-1)$

$$\underbrace{1}_{i=1} + \underbrace{n}_{i<n} + \underbrace{2}_{i++}\cdot(n-1) + \underbrace{4}_{\le 4}\cdot(n-1) = \boxed{7n-5}$$

for each value of j making $j<n$ true, $k<n$ evaluated $(n+1)$ times.

$k<n \rightarrow (F)$

j	k					
0	0	1	2	…	$n-1$	n
1	0	1	2	…	$n-1$	n
2	0	1	2	…	$n-1$	n
⋮			…			⋮
$n-1$	0	1	2	…	$n-1$	n
n						

$j<n \rightarrow (F)$

$$\underbrace{1}_{j=0} + \underbrace{n}_{k=0} + \underbrace{(n+1)}_{j<n} + \underbrace{n\cdot(n+1)}_{k<n} +$$

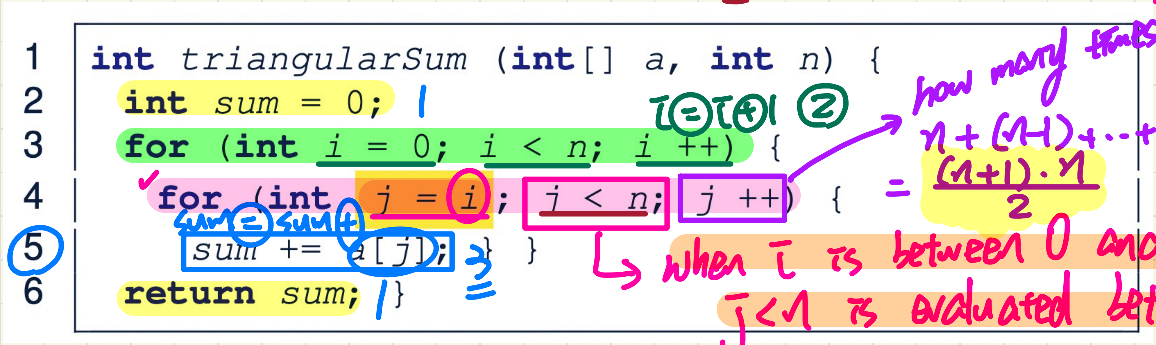
$$\underbrace{2\cdot n}_{j++} + \underbrace{2\cdot n^2}_{k++} + \underbrace{5\cdot n^2}_{\le 9} + \underbrace{1}_{\le 10}$$

$= ?$

Count # of Pos

$$24 + 21 + 18 + 15 + 12 = \frac{(24+12)*5}{2}$$

$$(n+1) + n + (n-1) + \dots + 2 = \frac{((n+1)+2)\cdot n}{2}$$



```
1  int triangularSum (int[] a, int n) {
2    int sum = 0;                         1
3    for (int i = 0; i < n; i ++) {       i=i+1  ②
4      for (int j = i; j < n; j ++) {
5        sum += a[j]; } }                 sum=sum+   3
6    return sum; }                        1
```

how many times?

$$n + (n-1) + \dots + 1 = \frac{(n+1)\cdot n}{2}$$

↳ when i is between 0 and n-1, j<n is evaluated between i and n.

$$\begin{array}{c|cccccc}
i & & & & & & \\
\hline
0 & 0 & 1 & 2 & \dots & (n-1) & n \\
1 & & 1 & 2 & \dots & (n-1) & n \\
2 & & & 2 & \dots & (n-1) & n \\
\vdots & & \vdots & & & & \\
n-1 & & & & & (n-1) & n
\end{array}$$

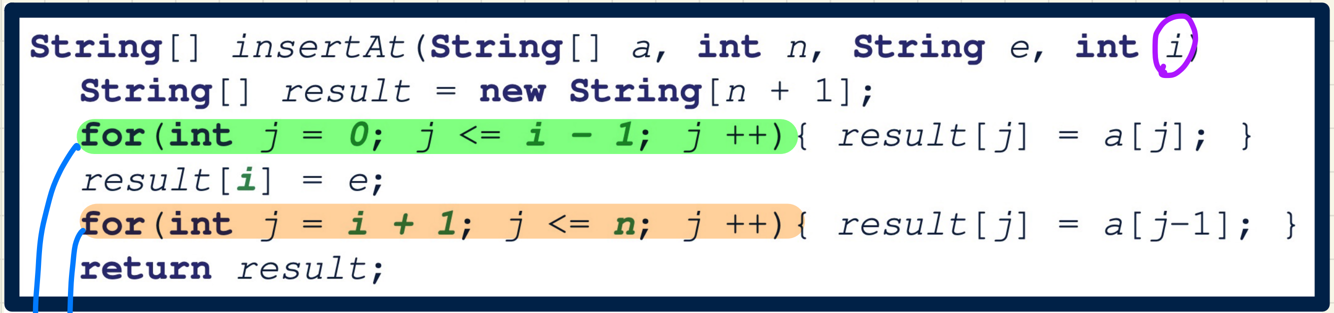
$j<n \rightarrow \textcircled{F}$

$i<n$ $\textcircled{n}$

$\downarrow$

$\textcircled{F}$

$$\underbrace{1}_{i=0} + \underbrace{n}_{j=i} + \underbrace{n+1}_{i<n} + \underbrace{\frac{((n+1)+2)\cdot n}{2}}_{j<n} + \underbrace{n\cdot 2}_{i++} + \underbrace{\frac{(n+1)\cdot n}{2}\cdot 2}_{j++} + \underbrace{\frac{(n+1)\cdot n}{2}\cdot 3}_{C5} + 1 = ?$$



```
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; j ++){ result[j] = a[j-1]; }
  return result;
```

```
for ( int j = 0 ; j <= i-1 ; j++) {
   for ( int k = i+1 ; k <= n ; k++) {

   }
}
```

j	k			
0	i+1	i+2	...	n
1	i+1	i+2	...	n
⋮				
i-1	i+1	i+2	...	n

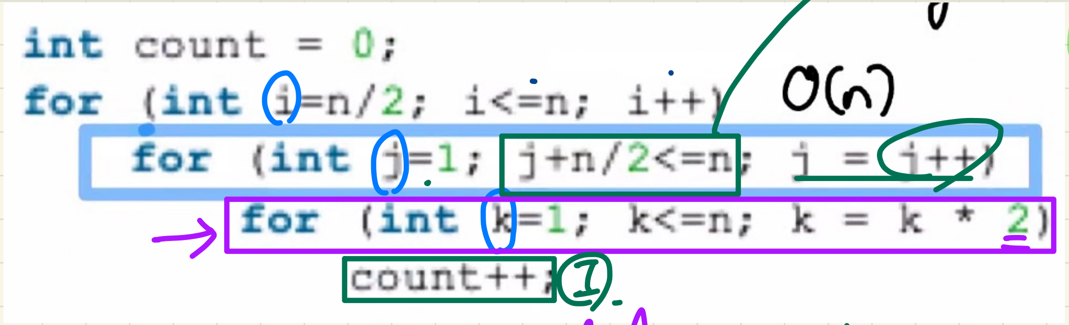
$n-(i+1)+1$

$= \boxed{n-i}$.

$(n-i)\cdot i$

$= \underline{n}\cdot\cancel{i} - i^2$ constant

$O(n)$



```
int count = 0;
for (int i=n/2; i<=n; i++)
    for (int j=1; j+n/2<=n; j = j++)
        for (int k=1; k<=n; k = k * 2)
            count++;
```

$O(n)$

$j + n/2 <= n$

$j <= n - \frac{n}{2} = \frac{n}{2}$

1

assume $n = 1000$

$k = 1 =$	2^0
$2 =$	2^1
$4 =$	2^2
$8 =$	2^3
⋮	
$512 =$	2^9

$10 = \lceil \log_2 1000 \rceil$

How many times j changes its value?

k: $1 \; \dots \; \log n \; \dots \; \frac{n}{2}$

I	J				
$n/2$	1	2	3	...	$\frac{n}{2}$
$n/2+1$	1	2	3	...	$\frac{n}{2}$
$n/2+2$	1	2	3	...	$\frac{n}{2}$
⋮					
n	1	2	3	...	$\frac{n}{2}$

$\approx \frac{n}{2}$

$O\left(\frac{n}{2} \cdot \frac{n}{2} \cdot \log n\right) = O(n^2 \cdot \log n)$