

## CS367 Announcements

### Thursday, June 27, 2013

- h1 graded
- h2 due Mon July 1st
- p1 due Wed July 3rd
- no class July 4th

#### Last Time

- Complexity (cont.)

#### Today

- Complexity (finish up)
- Primitive vs Reference

## Practice with analyzing complexity

Assume arrays **A** and **B** are each of size **N** (i.e.  $A.length=B.length=N$ )

### method1

```
public void method1(int[] A, int x, int y) {
    int temp = A[x];
    A[x] = A[y];
    A[y] = temp;
}
```

### method2

```
public void method2(int[] A, int s) {
    for (int i = s; i < A.length - 1; i++) {
        if (A[i] > A[i+1]) {
            method1(A, i, i+1);
        }
    }
}
```

## Practice with analyzing complexity

Assume arrays **A** and **B** are each of size **N** (i.e.  $A.length=B.length=N$ )

### method3

```
public void method3(int[] B) {
    for (int i = 0; i < B.length - 1; i++) {
        method2(B, i);
    }
}
```

### method4

```
public void method4(int N) {
    int sum = 0, M = 1000;
    for (int i = N; i > 0; i--) {
        for (int j = 0; j < M; j++) {
            sum += j;
        }
    }
}
```

## Practice with analyzing complexity

Assume arrays A and B are each of size N (i.e. A.length=B.length=N)

### method5

```
public void method5(int N) {
    int tmp, arr[];

    arr = new int[N];
    for (int i = 0; i < N; i++) {
        arr[i] = N - i;
    }

    for (int i = 0; i < N - 1; i++) {
        for (int j = i; j < N - 2; j++) {
            if (arr[j] > arr[j+1]) {
                tmp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = tmp;
            }
        }
    }
}
```

## Complexity caveats

## Primitive vs Reference Types: Assignment

### Primitive

```
int x, y, z;  
x = 7;  
y = x;  
z = x;  
y = 10;  
z = 8;
```

### Reference

```
ArrayList<String> x, y, z;  
x = new ArrayList<String>();  
y = x;  
z = x;  
y = new ArrayList<String>();  
z.add("Madison");
```