

CIS121 - Fall 2009

Lab 1 – Monday, September 13

Recall that when we analyze running time we analyze the *worst* case.

Count the exact number of steps for the following snips of pseudo-code and give a *tight* big-oh analysis

1. **Algorithm** *printSquare*(n)

Prints a square of '*' of size n

```
1  for  $i \leftarrow 0$  to  $n$ 
2      do for  $j \leftarrow 0$  to  $n$ 
3          do print *
4      end for
5      print lineBreak
6  end for
```

2. **Algorithm** *printTriangle*(n)

Prints a triangle of '*' of height n

```
1  for  $i \leftarrow 0$  to  $n$ 
2      do for  $j \leftarrow 0$  to  $i$ 
3          do print *
4      end for
5      print lineBreak
6  end for
```

3. **Algorithm** *countPositive*(A, n)

Count the number of positive integers in array A .

```
1   $count \leftarrow 0$ 
2  for  $i \leftarrow 0$  to  $n$ 
3      do if  $A[i] > 0$ 
4          then  $count \leftarrow count + 1$ 
5      end if
6  end for
7  print  $count$ 
```

4. **Algorithm** *weird*(A, n) Assume that the function $\min(a, b)$ is $O(1)$

```
1  count ← 0
2  for i ← 0 to n
3      do if  $A[i] \geq 0$ 
4          then count ← count + 1
5          else for j ← 0 to  $\min(A[i], n)$ 
6              do print *
7                  j ← j + 1
8          end for
9      end if
10 end for
```

5. **Algorithm** *weird2*(n) Assume that the function $\text{mod}(a, b)$ is $O(1)$

```
1  i ← 0
2  while i < n
3      do if  $\text{mod}(i, 2) = 0$ 
4          then for j ← 0 to n
5              do print *
6          end for
7      end if
8      i ← i + 1
9  end while
```

6. **Algorithm** *weird3*(n) Assume that the function $\text{mod}(a, b)$ is $O(1)$

```
1  i ← 1
2  while i < n
3      do if  $\text{mod}(i, 2) = 0$ 
4          then for j ← 0 to n
5              do print *
6          end for
7      end if
8      i ← i + 2
9  end while
```