

# CS61B SPRING 2016 SECRET SECTION 2 WORKSHEET

CS61B Tutors

Week 2

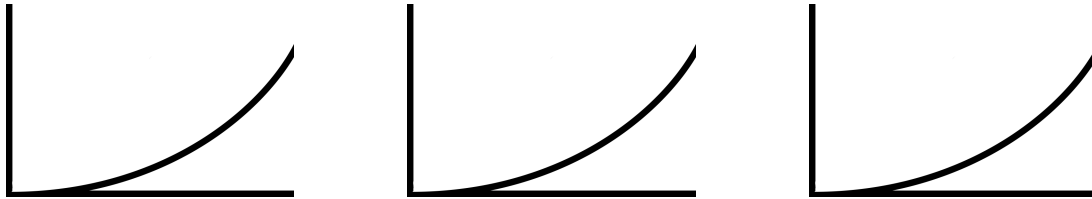
## 1 Big O Ordering

Rank the following from smallest to largest growths:

$O(\sqrt{n})$     $O(\log n)$     $O(2^n)$   
 $O(e^n)$     $O(n^{3/2})$     $O(n \log n)$   
 $O(1)$     $O(\log^2 n)$     $O(n!)$   
 $O(n^n)$     $O(n)$

## 2 Warmup

Starting from the graph on the left, shade in the regions that correspond to  $O(n^2)$ ,  $\Omega(n^2)$ ,  $\Theta(n^2)$ , respectively.



## 3 Big O Notation

Find the tightest  $O$ ,  $\Omega$ ,  $\Theta$  functions that bound the following:

1.  $5n + 6 - 3n$
2.  $2^n + 2^{n-1}$
3.  $n^2 + n \log n + 3n$
4.  $\log n + \log(n^2)$
5.  $\log n!$
6.  $1 + 2 + \dots + n$

## 4 Runtime Analysis

What are the  $O$ ,  $\Omega$ ,  $\Theta$  runtimes of the following function?

```
1 double minDistance = point[0].distance(point[1]);
2
3 /* Visit a pair (i, j) of points. */
4 for (int i = 0; i < numPoints; i++) {
5     /* We require that j > i so that each pair is visited only once. */
6     for (int j = i + 1; j < numPoints; j++) {
7         double thisDistance = point[i].distance(point[j]);
8         if (thisDistance < minDistance) {
9             minDistance = thisDistance;
10        }
11    }
12 }
```

## 5 More Runtime Analysis

What are the best case and worst case  $O$ ,  $\Omega$ ,  $\Theta$  runtimes of the following contrived function?

```
1 //runs in O(n) time
2 public static void linear(){...}
3 //runs in O(n^2) time
4 public static void squared(){...}
5 //runs in O(n^4) time
6 public static void fourth(){...}
7 //runs in O(n^5) time
8 public static void fifth(){...}
9
10 public static void contrived(n){
11     if (n % 2 == 0){
12         if (Math.random() > 0.5){
13             linear();
14         } else {
15             squared();
16         }
17     } else {
18         if (Math.random() > 0.5){
19             fourth();
20         } else {
21             fifth();
22         }
23     }
24 }
```

## 6 Even More Runtime Analysis

Assume `sortedList` is a sorted list of length  $n$  with no duplicates. What is the running time of the function `useless`? What does it print?

```
1  static void useless(int[] sortedList) {
2      for (int i = 0; i < sortedList.length; i++) {
3          System.out.println(foo(sortedList, sortedList[i]));
4      }
5  }
6
7  static int foo(int[] lst, int toFind) {
8      return bar(lst, toFind, 0, lst.length);
9  }
10
11 static int bar(int[] lst, int toFind, int lower, int upper) {
12     if (lower == upper) {
13         return -1;
14     }
15     int mid = (lower + upper) / 2;
16     if (lst[mid] > toFind) {
17         return bar(lst, toFind, lower, mid);
18     } else if (lst[mid] < toFind) {
19         return bar(lst, toFind, mid + 1, upper);
20     }
21     return mid;
22 }
```

## 7 Designing Algorithms

Write a function that determines if an array has all unique characters in  $O(n^2)$  time.

```
1 public static void hasUniqueCharacters(char[] characters){
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20 }
```

Now try to do it in  $O(n)$  time. Assume the only characters are lowercase a-z, 0-9.

```
1 public static void hasUniqueCharacters(char[] characters){
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20 }
```