

Name: \_\_\_\_\_ Section: \_\_\_\_ Andrew Id: \_\_\_\_\_

### 15-112 Fall 2014 Quiz 6

- \* 17½ minutes. No calculators, no notes, no books, no computers.
- \* You may not discuss any portion of this quiz with anyone until after 5pm today.
- \* SHOW YOUR WORK, CIRCLE YOUR ANSWERS.

1. **Short Answers** [20 pts; 5 pts each]

- 1) State and prove the worst-case big-oh runtime of selectionsort.
  
  
  
  
  
  
  
  
  
  
- 2) In the proof that mergesort is  $O(n \log n)$ , in just a few words, where does the  $\log n$  term come from?
  
  
  
  
  
  
  
  
  
  
- 3) For mergesort, if we multiply the input size by 4, we should expect the runtime to multiply by about what?
  
  
  
  
  
  
  
  
  
  
- 4) For some function  $f$ , it seems  $f(4n)$  runs about 32 times longer than  $f(n)$ . What is the probable big-oh of  $f$ ?

2. **Big-Oh** [30 pts; 5 pts each]: Indicate the worst-case big-oh runtime of each function. Show your work. For integer inputs, represent your answer in terms of  $n$ , not the number of bits needed to represent  $n$ . Assume all integers are non-negative and smaller than  $2^{64}$ , and so all arithmetic operations are constant  $O(1)$ .

```
def f1(n):  
    k = 1  
    while (k**2 < n): k += 1  
    return k
```

```
def f2(n):  
    k = 1  
    while (n > 0): (n, k) = (n/4, k+1)  
    return k
```

```
def f3(n):  
    k = 1  
    for i in xrange(n, n**2):  
        for j in xrange(n**3):  
            k += 1  
    return k
```

```
def f4(n):  
    k = 1  
    for i in xrange(n, n**2): k += 1  
    for j in xrange(n**3): k += 1  
    return k
```

### Big-Oh continued...

```
def f5(n):  
    k = 1  
    for val in xrange(3, n/4, 5): k += val  
    return k
```

```
def f6(n):  
    k = 1  
    for val in xrange(3, n/4, n/5): k += val  
    return k
```

### 3. Free Response: bubblesort [50 pts]

Write the function bubblesort, matching the design in the course notes. Be sure your function is  $O(n)$  when called on an already-sorted list. You may assume the function `swap(a, i, j)` is already written for you.

### 4. Bonus/Optional: [3 pts each]

1) Indicate what this will print:

```
def b(n):  
    def d(x): return sum([int(c) for c in bin(x).replace("b", "")])  
    return sum([x for x in xrange(n) if d(x) == 1])  
print b(100)
```

2) Rewrite the function `b2(n)` in a single line, so that it works the same for non-negative integer input.

```
def b2(n):  
    (s, result, a) = ("+-", 0, range(n))  
    for i in xrange(len(a)):  
        result = eval("%d%s%d" % (result, s[i%len(s)], a[i]))  
    return abs(result)
```