CSci 243 Homework 7

Due: 10:00 am, Wednesday, October 21

Alexander Powell

1. For each of the following program fragments, give an analysis of the running time. You may use summations to evaluate the running times of nested loops. Use big-O notation (i.e., ignore constants). Show your work.

   (a) (5 points)

```
for i = 1 to n
   for j = 1 to n
      if (i*j >= n)
         for k = 1 to n
             sum++
      endif
```

   Solution:

   The following program can be written in the pseudo-summation shown below:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\text{if } i\times j \geq n : \sum_{k=1}^{n} 1\right)$$

   which is equivalent to taking the entire summation and then subtracting the times when $i \times j < n$. This is shown below:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\text{if } i\times j \geq n : \sum_{k=1}^{n} 1\right) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} 1 - \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\text{if } i\times j < n : \sum_{k=1}^{n} 1\right)$$

   We can now rewrite this line as

$$n^3 - \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\text{if } i\times j < n : \sum_{k=1}^{n} 1\right)$$

   Because the second term in the previous line has a power of something less than three, it becomes insignificant as n grows towards ∞. Therefore, the n^3 term remains as the highest power so we can describe the run time complexity as $O(n^3)$.

(b) (5 points)

```
sum = 0
for i = 0 to n-1
   for j = 0 to i
      for k = 1 to n/(2^j)          (2^j) means 2 to the power j
         sum++
```

Solution:

We can write the program in the form of the triple sum shown below:

$$\sum_{i=0}^{n-1}\sum_{j=0}^{i}\sum_{k=1}^{\frac{n}{2^j}} 1 = \sum_{i=0}^{n-1}\sum_{j=0}^{i}\left(\frac{n}{2^j}\right)$$

From here we see the number of iterations that will be performed is triangular in a sense. That is, the number of iterations is represented as the sum $1+2+\ldots+n$, which is represented in the baby gauss formula $\frac{n(n+1)}{2}$, which is clearly of $O(n^2)$.

2. (10 points) Describe an algorithm that produces the maximum, the minimum, and the mean, of a set of n real numbers. Give an other algorithm that finds the median of this set. Give their time complexities.

Solution:

This first algorithm will produce the max, min, and mean of a dataset in the form of an array. Also, I'm assuming that the size of the set of numbers, n, is passed in as a parameter.

```
Algorithm1 (input: Array numbers[0] ... numbers[n-1], length n)
max = numbers[0]
min = numbers[0]
sum = 0
for i = 0 to n-1
    if (numbers[i] >= max)
        max = numbers[i]
    if (numbers[i] <= min)
        min = numbers[i]
    sum = sum + numbers[i]
mean = sum / n
// Output three values
return min, max and mean
```

This Algorithm linearly searches through the input of numbers, and keeps track of the biggest and smallest numbers it's seen so the time complexity is linear or $O(n)$.

The second algorithm returns the median of a dataset. This is done by sorting the set in ascending order and then depending on whether n is even or odd, returning the middle element or the average of the two middle terms.

```
Algorithm2 (input: Array numbers[0] ... numbers[n-1], length n)
// Begin by sorting the array
for i = 0 to n-1
    for j = i + 1 to n-1
        if (numbers[j] < numbers[i])
            tmp = numbers[j]
            numbers[j] = numbers[i]
            numbers[i] = tmp
// At this point the array is sorted, we just need to return median
if (n % 2 == 1) // if n is odd
    median = numbers[roundup(n/2) - 1]
if (n % 2 == 0) // if n is even
    median = numbers[n/2 - 1] + numbers[n/2]
    median = median / 2
return median
```

Because this algorithm is essentially a sorting algorithm with a couple calculations at the end, it will have the same time complexity as the sorting algorithm implemented. Since we used bubble sort, and it's worst case complexity is $O(n^2)$, so is Algorithm2.

3. (10 points) Use bubble sort to sort 8, 5, 1, 4, 3, 2, showing the lists obtained at each outer step.

   Solution:

   Using bubble sort, the given list is sorted as follows:

   $$8,5,1,4,3,2$$

   $$5,1,4,3,2,8$$

   $$1,4,3,2,5,8$$

   $$1,3,2,4,5,8$$

   $$1,2,3,4,5,8$$

   $$1,2,3,4,5,8$$

4. (10 points) Use insertion sort to sort 8, 5, 1, 4, 3, 2, showing the lists obtained at each outer step.

   Solution:

   Using insertion sort, the given list is sorted as follows:

   $$8,5,1,4,3,2$$

   $$5,8,1,4,3,2$$

   $$1,5,8,4,3,2$$

   $$1,4,5,8,3,2$$

   $$1,3,4,5,8,2$$

   $$1,2,3,4,5,8$$