Taras Derewecki

CMPSC 463 Problem Set #5 (100 points)

1. Consider the partition algorithm from quicksort. (30 points)

a) Using this algorithm, write a method that finds the kth largest element in an array of integers. This element is defined as with minimum value out of all elements whose value is greater than or equal to at least k-1 elements in the array.

```
int partition(int arr[], int l, int r);

int kthLargest(int arr[], int l, int r, int k){
    if (k > 0 && k <= r - l + 1){
        int position = partition(arr, l, r);
        if (position-l == k-1)
            return arr[position];
        if (position-l > k-1)
            return kthLargest(arr, l, position-1, k);

        return kthLargest(arr, position+1, r, k-position+l-1);
    }
}
```

b) Suppose that your algorithm always picks a "good enough" pivot, defined as one that reduces the size of the array that contains the desired element by at least 3/4. Write a recurrence relation that expresses the worst case running time of your algorithm.

The "good enough pivot" reduces the size of the array by 3/4 i.e, we only have to recurse through n/4.

⇨ **T(n) = T(n/4) + O(n)(partition is O(n))**

c) Solve the recurrence relation.

Using Master's Theorem:
T(n) = T(n/4) + O(n)
a = 1, b = 4, f(n) = O(n) (c = 1), h(n) = $\log_4(1)$ => c > h(n)
=> 1 > 0
T(n) = O(f(n)) = O(n)

d) If we can choose a "good enough pivot" at least half of the time, what would be the average case running time of your algorithm?

Analogous to quicksort, the average time complexity in this case would be O(n log(n))

Taras Derewecki

2. The *One Size Fits One* t-shirt company that makes unique sizes for its customers. Unfortunately, in the latest batch of t-shirts, the customer assignments for each t-shirt have been lost. Moreover, it is impossible to sort the shirts (or the people) just by looking at them. The customers have been invited to try on the shirts to try to figure out how to assign shirts. When a customer tries on a shirt, you will be only tell if the shirt is too small, fits, or is too large.

Your task is to write two methods in Java to help with the assignment given *n* unique customers and *n* unique shirts. You should use the following template when writing your code.

a) The first method should be an $\theta(n^2)$ expected time algorithm that rearranges the people and/or the shirts such that such that persons.get(i) fits shirts.get(i) for all 1 <= i <= n. (30 points)

b) The second method should be a θ(n log(n)) expected time algorithm that rearranges the nuts and/or the bolts such that such that persons.get(i) fits shirts.get(i) for all 1 <= i <= n. (40 points)

The algorithms should use the classes in the template, but you may only call the `equals` (aka fits) and `isTooSmall` methods in your algorithm, i.e. you may not access the protected size variables.

Question 2 part A and part B are both answered in the java file submitted with this pdf file.

Please submit your answer to problem 1 as a pdf file, and your answer to problem 2 as a .java file. You may change the methodA and methodB methods, and add methods to the Solution class, but otherwise the template code should be left unchanged.