Analysis of Algorithms

Sign the exam with your student number - not your name

Answer all three questions to the best of your ability.

1. (40 pts) Provide a time and space analysis of the shaker sort algorithm below. (author
Jason Harrison@cs.ubc.ca, with modifications by wds@cs.fit.edu 11/7/00)

```java
1   (Shaker Sort 1)
    void shakerSort(int a[]) throws Exception {
        int i = 0; int k = a.length- 1;
        while (i < k) {
            int min = i; int max = i; int j;
            for (j = i + 1; j <= k; j++) {
                if (a[j] < a[min]) { min = j; }
                if (a[j] > a[max]) { max = j; }
            }
            int T = a[min]; a[min] = a[i]; a[i] = T;
            if (max == i) {
                T = a[min]; a[min] = a[k]; a[k] = T;
            } else {
                T = a[max]; a[max] = a[k]; a[k] = T;
            }
            i++; k--;
        }
    }
```
2. (40 pts) Consider the following algorithm that finds the maximum element in an array $A[0...n - 1]$. Find a recurrence relation (and initial condition) that characterizes the running time complexity of the algorithm. Solve the recurrence you found.

```java
public int maximum(int[] A, int lo, int hi) {
    if (hi - lo <= 1) {
    }
    else {
        int max1 = maximum(A, lo, (lo + hi)/2);
        int max2 = maximum(A, (lo + hi)/2 + 1, hi);
        return (max1 < max2) ? max2 : max1;
    }
}
```
3. (20 pts) A little problem: “Rotate an array of $N$ elements left by $I$ positions.” For example, with $N = 8$ and $I = 3$, the array $ABCDEFGH$ is rotated to $DEFGHABC$.

(a) Write a simple program (in the language of your choice) to do the job.
(b) What is the time and space complexity of your solution?
(c) Can you show how to rotate the array in time proportional to $N$ in constants space?