

Homework 8

Due 09/30/16

September 27, 2016

1. Find a recurrence that describes the worst-case complexity of the following recursive sorting algorithm. Show all work. You may assume that the floor function ($\lfloor \cdot \rfloor$) takes constant time.

Input: *data*: an array of integers
Input: *n*: the length of *data*
Output: a permutation of *data* such that
 $data[1] \leq data[2] \leq \dots \leq data[n]$

```
1 Algorithm: ThirdSort
2 if  $n = 1$  then
3   | return data
4 else if  $n = 2$  then
5   | if  $data[1] > data[2]$  then
6   |   | Swap  $data[1]$  and  $data[2]$ 
7   | end
8   | return data
9 else
10  |  $third = \lfloor n/3 \rfloor$ 
11  | Call ThirdSort on  $data[1..n-third]$ 
12  | Call ThirdSort on  $data[third+1..n]$ 
13  | Call ThirdSort on  $data[1..n-third]$ 
14  | return data
15 end
```

2. Use the Master Theorem to find the worst-case complexity of ThirdSort and describe how ThirdSort compares to SelectionSort.

You may assume that $f(n)$ is regular if relevant. Recall that $\log_a(b) = \frac{\ln(b)}{\ln(a)}$ (you may need a calculator for this one). Be sure to include the value of c and the case of the Master Theorem in your answer.