# CS 581 Homework 4

February 11, 2018

## Problems (Due on 02/08/2018)

**Problem 1.** A previous CS581 student (let's call him Jerome) took this class and proposed the following sorting algorithm:

```
JSORT(LIST)
if LIST.length=0,1 then return LIST
if LIST.length=2 then compare items, swap if necessary and return LIST
JSORT the leftmost two-thirds of LIST
JSORT the rightmost two-thirds of LIST
JSORT the leftmost two-thirds of LIST
return LIST
```

a) Argue convincingly either that JSORT does or that it does not work.
b) In any case, determine JSORT's time complexity.

**Problem 2.**
Prove that the running time of building a heap is $O(n)$. You can assume a MAX-heap.

**Problem 3.**
Determine if array $\{27, 17, 3, 16, 13, 10, 1, 5, 7, 12, 4, 8, 9, 0\}$ is a MAX-heap. If not, illustrate the operation of Max-heapify by sinking or floating down nodes. This assumes the storage scheme described in the textbook.

**Problem 4.** You are given the job of partitioning $2n$ players into two teams of $n$ players each. Each player has a numerical rating that measures how good he/she is at the game. Newt seeks to divide the players as unfairly as possible, so as to create the biggest possible talent imbalance between team A and team B. Show how the job can be done in $O(n \log n)$ time.

**Problem 5.** Suppose an array A consists of $n$ elements, each of which is red, white, or blue. We seek to sort the elements so that all the reds come before all the whites, which come before all the blues. The only operation permitted on the keys are

- Examine$(A, i)$: report the color of the $i$th element of $A$.
- Swap$(A, i, j)$: swap the $i$th element of $A$ with the $j$th element.

Find, describe and analyze a linear time algorithm for this red-white-blue sorting.

# Programming Assignment (Due on 02/15/2018)

**Problem 1.** Recall that in the selection problem, we take as input a list $L$, its length $n$, and an integer $k$. We are asked to report the value of $L$'s $k$th smallest element. In class we discussed the median-of-medians rule. Use C99 or C++98 to implement this technique and solve the selection problem in linear time. The code should compile and run on the Hydra Lab machines. Specifically,

- Set $r$ at one of 5, 7, 9 or 11. Justify your decision.
- Choose your sorting algorithm when n becomes small and justify the decision.
- Experimentally determine the value of n at which your linear-time selection code should revert to sorting and unwind the recursion. Describe how you made this determination, and the factors upon which it depended.

In addition,

- Suppose your code is compiled to executable program `linear_select`. It should take command line arguments as

  ```
  linear_select input_file k n
  ```

  where input_file is the name of a file containing integers separated by spaces. Your program should read in the first n elements in that file and output the $k$th smallest element among them.
- No need to do any error-checking of inputs.
- The possible n values will be 1024 (1k) to 1024*1024 (1M), and the input file will have sufficient elements in it.
- Include a README file if your program needs special compile procedures or parameter passing methods.

Submission: Send the TA all files necessary to compile and run your code before it's due.