Fundamentals of Computer Algorithms
Homework 2 Additional Problems

Prof. Matthew Moore

Due: 2018-09-04

1. In the python file associated with this homework, implement the functions

    1. `Heap._parent_index`,
    2. `Heap._parent`,
    3. `Heap._parent_key`,
    4. `Heap._heapify_up`.

    Follow the instructions in the source file. In your printed homework submission, include *only* these functions, do not print the entire source file.

2. In the python file associated with this homework, implement the functions

    1. `PQ.add`,
    2. `PQ.pop`.

    Follow the instructions in the source file. In your printed homework submission, include *only* these functions, do not print the entire source file.

3. In the python file associated with this homework, implement the function `sort_with_PQ`.

    Your solution should be linear in the priority queue operations. That is, it should run in $\mathcal{O}(n \cdot \mathcal{O}(PQ))$, where $\mathcal{O}(PQ)$ is the $\mathcal{O}$-complexity of the priority queue operations. Since the heap operations are all $\mathcal{O}(\log(n))$, this sort algorithm is $\mathcal{O}(n \log(n))$. It is called *heapsort*, and is used by default in the Linux kernel due to its more stable performance characteristics.

4. Let L be a list and define property (A) on elements of L by

    $$L[i] \text{ is not smaller than its neighbors, if they exist.} \tag{A}$$

    "Neighbors" refers to the elements $L[i-1]$ and $L[i+1]$ if they exist. Here is an algorithm that will find *one* element of L exhibiting property (A).

    (i) What is the Θ-complexity of `findA`?

    (ii) Implement a function `better_findA` that performs better than `findA`. That is, if $f(n)$ and $b(n)$ are the number of iterations of `findA` and `better_findA`, respectively, then

    $$\lim_{n\to\infty} \frac{b(n)}{f(n)} = 0.$$

5. Suppose we have two parameters, m and n, with $m \to \infty$ and $n \to \infty$, perhaps at different rates independent of one another. Which has larger Θ-complexity: $m^{\ln(n)}$ or $n^{\ln(m)}$?