# CS5800: Algorithms — Virgil Pavlu

Homework 8

Name: Harsh Shah
Collaborators:

Instructions:

- Make sure to put your name on the first page. If you are using the LaTeX template we provided, then you can make sure it appears by filling in the `yourname` command.
- Please review the grading policy outlined in the course information page.
- You must also write down with whom you worked on the assignment. If this changes from problem to problem, then you should write down this information separately with each problem.
- Problem numbers (like Exercise 3.1-1) are corresponding to CLRS $3^{rd}$ edition. While the $2^{nd}$ edition has similar problems with similar numbers, the actual exercises and their solutions are different, so make sure you are using the $3^{rd}$ edition.

**1.**

16.3.3
Consider an ordinary binary min-heap data structure supporting the instructions INSERT and EXTRACT-MIN that, when there are n items in the heap, implements each operation in $O(\log n)$ and the amortized cost of EXTRACT-MIN is O(1), and show that your potential function yields these amortized time bounds. Note that in the analysis, n is the number of items currently in the heap, and you do not know a bound on the maximum number of items that can ever be stored in the heap.

**Solution:**
Suppose f(n) is the runtime of the EXTRACT-MIN operation on a heap with n items. If we define $\phi = \sum_{k=1}^{n} f(k)$, then the potential drops by f(n) when an EXTRACT-MIN operation is performed. Thus, the cost incurred by an EXTRACT-MIN operation is exactly cancelled out by the change in potential and the amortized cost of the operation is O(1). Furthermore, it is okay to increase the potential by f(n) on each INSERT operation since $f(n) = O(\log n)$ and thus the amortized cost of the INSERT operation is $O(\log n) + O(\log n) = O(\log n)$.
Formally, define $\phi = \sum_{k=1}^{n} f(k)$, where n is the number of items in the heap after the i-th operation. We have $\phi_0 = 0$ and $\phi_i \geq 0$ for all i since f is a non-negative function (it is the number of steps performed by an algorithm). Also, if the i-th operation is INSERT, then $c_i' = c_i + \Delta\phi_i = \log n + f(n) = O(\log n)$. If the i-th operation is an EXTRACT-MIN operation we have $c_i' = c_i + \Delta\phi_i = f(n) - f(n) = 0 = O(1)$ which is what is needed.