Problems: Amortized Analysis

Palash Dey
Indian Institute of Technology, Kharagpur

July 31, 2024

1. [CLRS book] Recall the dynamic table data structure discussed in the class. Suppose that instead of contracting a table by halving its size when its load factor drops below $1/4$, we contract it by multiplying its size by $2/3$ when its load factor drops below $1/3$. Show that insert and delete operations take $\mathcal{O}(1)$ amortized time. f(i) = 2.num - size, works for both cases.

2. [CLRS book] Design a data structure to support the following two operations for a dynamic multiset S of integers, which allows duplicate values:

   INSERT(S,x) inserts x into S.

   DELETE-LARGER-HALF(S) deletes the largest $\lceil |S|/2 \rceil$ elements from S.

   Explain how to implement this data structure so that the amortized time complexity of INSERT and DELETE-LARGER-HALF is $\mathcal{O}(1)$.

3. [CLRS book] Consider an ordinary binary min-heap data structure with n elements supporting the instructions INSERT and EXTRACT-MIN in $\mathcal{O}(\log n)$ worst-case time. Show that the amortized cost of INSERT is $\mathcal{O}(\log n)$ and EXTRACT-MIN in $\mathcal{O}(1)$. f(i) = ilog(i)