Illinois Institute of Technology
Department of Computer Science

Homework Assignment 3

CS 535 Design and Analysis of Algorithms
Fall Semester, 2015

Rules for Homework

Remember, the rules listed on the first homework assignment apply to all assignments.

Due: Thursday, September 17, 2015

1. Suppose in the example of expansion/contraction in hash tables (September 10 lecture; section 17.4.2 of CLRS3) we change the potential function to

$$\Phi_i = \begin{cases} 2 \cdot num_i - size_i & \text{if } \alpha_i \geq 1/2, \\ size_i - 2 \cdot num_i & \text{if } \alpha_i < 1/2. \end{cases}$$

   Go through all 8 cases (as per the lecture) and calculate the amortized costs. Does the analysis still work—that is, give amortized $O(1)$ insertion/deletion costs?

2. We want a data structure that supports three operations on a list of records (we know nothing about the structure or content of the records):

   Insert(x, y): Insert the new record y as the successor to record x.
   Delete(x): Remove record x from the list.
   Order(x, y): Return **true** if x precedes y in the list; otherwise, return **false**.

   We use a data structure consisting of a circular, doubly-linked list of the records with header record H. In addition to the contents of the records, we maintain for each record r a non-negative integer label(r) and pointers to the successor record succ(r) and the predecessor record pred(r). The integers available for labeling the records are $0, 1, 2, \ldots, M-1$, where M is some fixed value such that fewer than $\sqrt{M}/2$ item are ever stored in the list.

   The initial (empty) data structure consists only of the header record H which has label 0; its predecessor and successor pointers point to itself. We define the label of the successor of a node r,

$$V(r) = \begin{cases} M & \text{if succ}(r) = H, \\ \text{label}(\text{succ}(r)) & \text{otherwise,} \end{cases}$$

   and it will always be true that

$$\text{label}(r) < V(r). \tag{1}$$

   The three operations are done as follows:

   Order(x, y): Return **true** if label$(x) <$ label(y); otherwise, return **false**.
   Delete(x): Delete x from the circular list.

INSERT(x, y): If $\text{label}(x) + 1 = V(x)$, do the RELABEL operation (below). Now $\text{label}(x) + 1 < V(x)$ and we insert y between x and $\text{succ}(x)$ and assign it the label $\lfloor(\text{label}(x) + V(x))/2\rfloor$; this maintains the invariant (1).

The RELABEL operation is as follows:

Let n be the number of records in the list before the insertion and let

$$w_i = \begin{cases} [V(\text{succ}^{(i-1)}(x)) - \text{label}(x)] \bmod M & \text{if } 0 \le i < n, \\ M & \text{if } i = n. \end{cases}$$

Run the following algorithm to compute j:

$i := 1$; $j := 2$;
while $w_j \le 4w_i$ **do**
  $i := i + 1$; $j := \min(2i, n)$;
end while

Now, relabel the $j - 1$ records $\text{succ}^{(1)}(x), \ldots, \text{succ}^{(j-1)}(x)$ with the labels

$$\text{label}(\text{succ}^{(k)}(x)) = \lfloor w_j k/j \rfloor + \text{label}(\text{succ}^{(i)}(x)) \bmod M.$$

Using the potential function

$$\Phi = \sum_{0 \le k < n} -\log g_k,$$

where

$$g_k = \text{label}(\text{succ}^{(k+1)}(H)) - \text{label}(\text{succ}^{(k)}(H))$$

is the size of the gap between successive labels do the following:

(a) Prove that the amortized cost of ORDER is $O(1)$.

(b) Prove that the amortized cost of DELETE is $O(1)$.

(c) Prove that the amortized cost of INSERT, including any relabeling, is $O(\log M)$.

(d) Explain why the number of items stored in the list must be fewer than $\sqrt{M}/2$.