Illinois Institute of Technology
Department of Computer Science

Homework Assignment 3

CS 535 Design and Analysis of Algorithms
Fall Semester, 2016

Due: Thursday, September 15, 2016

Remember the Honesty Pledge!

1. Do problem 17.3-3 on page 462 of CLRS. (*Hint*: Consider a potential function proportional to the sum of the depths of the nodes in the heap.)

2. We want to maintain a collection of k dynamic tables in a sequential segment of memory so that the amortized cost of an insertion or deletion is $O(k)$, generalizing Section 17.4 of the text.

   Let the set of tables $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$ be sequentially arranged in memory, with $\texttt{size}(T_i)$ locations allocated to the $\texttt{num}(T_i)$ elements stored in T_i, so that T_i has $\texttt{size}(T_i) - \texttt{num}(T_i)$ vacant slots. Let S and N denote the total numbers of slots and filled slots, respectively, in all the tables in $\mathcal{T}$. Thus, the tables contain $S = \sum_{i=1}^{k} \texttt{size}(T_i)$ contiguous slots, of which $N = \sum_{i=1}^{k} \texttt{num}(T_i)$ are occupied. Assume that the cost of relocating the set of tables $\mathcal{T}$ is $O(N)$. Let T_i, S, and N denote the tables/values *before* the operation under consideration, and the corresponding primed symbols T_i', S', and N' denote the tables/values *after* the operation.

   Initially, all the tables are empty and no space is allocated for them; that is, $\texttt{size}(T_i) = \texttt{num}(T_i) = 0$, for $1 \le i \le k$, and hence $S = N = 0$. If $\texttt{size}(T_i) > \texttt{num}(T_i)$ immediately before an insertion to table T_i, there is room in the table and the new element is inserted without difficulty. But if $\texttt{size}(T_i) = \texttt{num}(T_i)$, the table is full and cannot accommodate the new element. In this case, *all* the tables are copied into a larger area, increasing the size of T_i by $N + 2$ in the process; the new element is then inserted into the enlarged table. Thus, immediately after such an insertion, $\texttt{size}(T_i') = \texttt{num}(T_i') + N'$.

   To prevent tables from becoming too sparse, each deletion includes an examination of the sizes of *all* the tables, and the contraction of any table T_i for which $\texttt{size}(T_i') > \texttt{num}(T_i') + 2N'$. When deleting an entry from T_i, if $\texttt{num}(T_i) > \texttt{size}(T_i) - 2N + 2$ immediately before the deletion, then the element is simply deleted. However, if $\texttt{num}(T_i) \le \texttt{size}(T_i) - 2N + 2$, the element is deleted and then T_i is contracted to size $\texttt{num}(T_i) + N - 2$ by copying the elements in all the tables into a smaller region. In either case, the deletion decrements the value of N; consequently, any other table T_i with $\texttt{size}(T_i) - 2N \le \texttt{num}(T_i) < \texttt{size}(T_i) - 2N + 2$ before the deletion will be contracted during such a deletion. Any table T_i that has been contracted satisfies $\texttt{size}(T_i') = \texttt{num}(T_i') + N'$ immediately after the contraction. Notice that a single deletion can result in the contraction of many tables, which may or may not include the table in which the deletion actually occurred. This contrasts with insertion: an insertion cannot result in the expansion of any table other than the one in which the insertion actually occurred.

   Define the potential of $\mathcal{T}$ as

   $$\Phi(\mathcal{T}) = \sum_{i=1}^{k} |\, \texttt{size}(T_i) - \texttt{num}(T_i) - N|.$$

   (a) Prove that the contribution of table T_i to $\Phi(\mathcal{T})$ is zero immediately after it expands or contracts.

(b) Prove that the amortized cost of an insertion that does not require an expansion is at most $k+O(1)$.

(c) Prove that the amortized cost of an insertion that requires an expansion is $O(k)$.

(d) Prove that the amortized cost of a deletion is $O(k)$. (*Hint*: There are three cases—no contraction, the table in which the deletion occurs is also contracted, and some tables contract but the table that is the site of the deletion is not one of them.)

(e) Give a sequence of m insertions/deletions that requires time $\Omega(km)$.

(f) The *load factor* $\alpha(\mathcal{T}) = N/S$ is the fraction of the allocated memory in use. Prove all tables always satisfy

$$\texttt{size}(T_i) \leq \texttt{num}(T_i) + 2N,$$

and use this to prove that

$$\alpha(\mathcal{T}) \geq \frac{1}{2k+1}.$$

3. **PhD Qualifying Exam Section Problem 3.**

Do problem 17-4 on pages 474–475 of CLRS.