CSci 231 Homework 8 [*]

Amortized Analysis

CLRS Chapter 17

Use a (single) separate sheet of paper for each problem. Be concise.

1. (CLRS 17.2-1) A sequence of stack operations is performed on a stack whose size never exceeds k. After every k operations, a copy of the entire stack is made for backup purposes. Show that the cost of n stack operations, including copying the stack, is $O(n)$ by assigning suitable amortized costs to the various stack operations.

2. A sequence of n operations is performed on a data structure. The ith operation costs i if i is a power of 2, and 1 otherwise. Using the accounting method, determine the amortized cost per operation.

3. In this problem we consider two stacks A and B manipulated using the following operations (n denotes the size of A and m the size of B):

    - $PushA(x)$: Push element x on stack A.
    - $PushB(x)$: Push element x on stack B.
    - $MultiPopA(k)$: Pop $\min\{k, n\}$ elements from A.
    - $MultiPopB(k)$: Pop $\min\{k, m\}$ elements from B.
    - $Transfer(k)$: Repeatedly pop an element from A and push it on B, until either k elements have been moved or A is empty.

    Assume that A and B are implemented using doubly-linked lists such that *PushA* and *PushB*, as well as a single pop from A or B, can be performed in $O(1)$ time worst-case.

    (a) What is the worst-case running time of the operations *MultiPopA*, *MultiPopB* and *Transfer*?

    (b) Define a potential function $\Phi(n, m)$ and use it to prove that the operations have amortized running time $O(1)$.

4. CLRS 17-2 page 426

[*] Collaboration is allowed and encouraged, if it is constructive and helps you study better. Remember, exams will be individual. List the names of the collaborators with the solutions.

5. EXTRA CREDIT: (CLRS 17.3-5) Suppose that a counter begins at a number with b 1's in its binary representation, rather than at 0. Show that the cost of performing n INCREMENT operations is $O(n)$ if $n = \Omega(b)$. (Do not assume that b is constant.)