

CS 840 Assignment 1  
Due October 13, 2016, in class  
Instructor: I. Munro

1. Consider the idea of using a hash function in the van Emde Boas “priority queue” to reduce the space requirements. Assume that hashing a key to a location takes  $O(1)$  time and space for the hash table is proportional to the size of the data entries and keys. Take the scheme for hashing as a “black box”. How can you use such hashing to implement the van Emde Boas method efficiently?
  - a. Outline your scheme, perhaps by showing how it differs from the standard version.
  - b. Assuming the hashing works in constant time for an insert, delete or search, how much time is taken for each of the operations: insert, delete and successor.
  - c. What is the space requirement of your method? Assume you have a universe of  $u$  elements and a word size of  $\lg u$  bits.  $n$  elements are currently stored in the data structure.
2. We have discussed, in great detail, the Move to Front heuristic for improving linear search. This question deals with the very simple notion of counting accesses to elements and maintaining the list in order by Frequency Count (FC).
  - a. Outline a method for keeping frequency counts and maintaining the elements of the list in decreasing order by their counts. This method should require only  $O(1)$  time to update a count and to restore and to move such an updated element to its appropriate place in the list.
  - b. If elements are drawn from a fixed (and independent) distribution, it can be shown that the expected behaviour of FC is close to that of the optimal. You don't have to prove this. We will concentrate on the amortized (worst case) behaviour, and compare FC with (i) the static optimal ( $S_{opt}$ ) which keeps elements in decreasing order of frequency of the entire sequence and (ii) the dynamic offline optimal.
    - i. Prove that the amortized case behaviour of the FC heuristic is within a constant factor of that of  $S_{opt}$ . (As in class, assume FC starts with an empty list and adds an element to the end of the list when it is first accessed)
    - ii. Consider the model of Sleator and Tarjan in which a list may be rearranged by moving the requested element toward the front of the list at no charge, or swapping two adjacent elements at a cost of 1. Prove that under this model, FC is not necessarily within a constant factor of the offline optimal on an amortized basis.