Computer Science 3711 / Engineering 5891 (Winter 2002):
Assignment #3
Due: March 14, 2002

1. (40 marks) Implement (in C, C++, or Java) the following basic set of operations for a binary min-heap:

   • MIN-HEAPIFY(A, 1)
   • BUILD-MIN-HEAP(A)
   • HEAP-MINIMUM(A)
   • HEAP-EXTRACT-MIN(A)
   • HEAP-DECREASE-KEY(A,i,key)
   • MIN-HEAP-INSERT(A, key)

The algorithms for all of these operations are modifications of those given for max-heaps in Chapter 6 of the textbook. Write a main program that uses these operations to create a min-priority queue that stores (key, symbol) pairs (you may find it convenient to add a symbol-“handle” parameter to the appropriate operations above in order to do this). This program will read in a sequence of such pairs, store them in the min-priority queue, and then remove the keys in increasing value from the min-priority queue while printing their associated keys in that order. For example, given the key-symbol sequence (10, ”D”), (5, ”B”), (4, ”A”), (8, ”C”)), the main program will print out

   key-symbol pair # 1: ( 4,"A")
   key-symbol pair # 2: ( 5,"B")
   key-symbol pair # 3: ( 8,"C")
   key-symbol pair # 4: (10,"D")

Run your main program on the following test cases:

     (21,"t"),(12, "y"),(11, ","),(16, ","),(7, ","),(152, "!"))
   • ((12,"a"),(206,"e"),(202,"e"),(200,"h"),(4,"h"),(6,"i"),(204,"r"),
     (18,"t"),(10,"y"),(2, ","),(16, ","),(210, ","),(14, ","))
   • ((144,"a"),(21,"e"),(8,"e"),(1,"h"),(5,"h"),(1,"i"),(13,"r"),
     (3,"t"),(89,"y"),(55, ","),(2, ","),(253, ","),(34, ","))

Sequences may read in one key-symbol pair per line. Hand in your code and copies of your output on these test cases.

2. (20 marks) Derive a dynamic programming algorithm for finding the longest monotonically increasing subsequence of a sequence of n integers. In a monotonically increasing sequence of integers (a1, a2, ... , an), ai < ai+1 for 1 ≤ i < n. For example, the longest monotonically
increasing subsequence of the sequence (2, 10, 8, 1, 9, 3, 6) has length 3, and several such sub-
sequences are (2, 8, 9), (2, 3, 6), and (1, 3, 6). Give the recurrence relation (base and recursive
cases), the table fill-in procedure, the traceback procedure for finding an optimal subsequence,
and the time and space complexities of both procedures.

3. (10 marks) Show how to compute the length of a longest common subsequence using only
2\min(m,n) entries in the \(c\) table plus \(O(1)\) additional space. Then, show how to do this
using \(\min(m,n)\) entries plus \(O(1)\) additional space. Algorithms are not necessary; informal
arguments and diagrams will suffice for an answer.

4. (20 marks) Consider the following problem:

Bob Fish wants to drive his truck from St. John’s to Port Aux Basques along the
Trans Canada Highway. His truck’s gas tank, when full, holds enough gas to travel
\(n\) miles, and his map gives the distances between gas stations on his route. Bob
wishes to make as few gas stops as possible along the way.

Derive a greedy algorithm for solving this problem. Give the algorithm itself, a proof that
the algorithm is correct, and the time and space complexity of the algorithm.

5. (10 marks) Many greedy approaches to the activity selection problem (Section 16.1, Text-
book) do not always produce a maximum-size set of mutually compatible activities. For each
of the following approaches, give an example (that is, a set of activities to be scheduled) for
which that approach fails to produce an optimal solution:

(a) Selecting the compatible remaining activity of least duration.
(b) Selecting the compatible remaining activity with the earliest start time.