COMP 5511/2 Bipin C. Desai Fall 2023
Assignment 3. Due. as in CrsMgr
Make sure to do a peer evaluation ASAP.

Theoretical Component - Each question is worth 10 points

1. Suppose the internal nodes of two binary trees T1 and T2 respectively, hold items that satisfy the heap-order property. Describe a method for combining these two trees into a tree T, whose internal nodes hold the union of the items in T1 and T2 and also satisfy the heap-order property. Note that although the items of the trees (T, T1 and T2) satisfy the heap-order property, they are not necessarily heaps (i.e., they are not necessarily complete trees). Your algorithm should run in O(h1 + h2) time where h1 and h2 are the respective heights of T1 and T2.
2. Draw the hash table of size 17 that results from using the hash function h(i) = (2i + 5)mod 17; to hash the keys 12, 44, 13, 17, 24, 88, 23, 94, 11, 39, 34, 20, 16, and 51, assuming collisions are handled (a) by chaining (b) by linear probing in a decreasing direction (c) by double hashing using a secondary hash function h2(i) = 7−(k mod 7).
3. Create the optimum Hoffman code using the following words. YOU, ME, WE, SHE, HE, SOW, COW, DOG, PIG, RIG, GOLD, SEA, RUG, HAT, CAT,ROW, MOB, LOG, BOX, TAB, BAR, EAR, TAR, JAR, DIG, FAN, BIG, TEA, NOW, FOX, BOG, BAT, BIT, KIT, SIT, ZEN, RAN, FAN, QUIZ, VAN

- Include the comma(,) and the space as part of your set of ’alphabet’! For standardization, place the lower frequency node or subtree on the left which would be the 0 branch.
- Thence using this give the coding for:
    - YOU GOT TO HAVE FUN BUT YOU NEED TO KNOW HOW SAID THE CAT IN THE HAT

Programming Components

1. This part is worth 40 Points Consider a record which contains two keys; these keys may be unique (e.g. in employee information the unique keys could be Employee ID and Social Insurance Number(SIN) or for a bank customer, it could be the Account number and the persons SIN) or only one may be unique(e.g. A book has a ISBN (unique) and a Title(not necessarily unique, but assume to be unique if combined with the name of the Author).

Write a program that reads a file text containing a number of records. The first record has the names of the fields (the first two fields being the ones on which the record is to be sorted) and the subsequent records has the data for it. The end of data is indicated by the EOF.

Your program would read the data(created by you, but it could be used by the marker to add or delete records) and create appropriate ADT which would then be used to store and sort the records using the heap sort method based on the first field and thence sort it in place. It would then create a binary tree for help in the search of the appropriate record based on the second key(which may or may not be unique).

Employee record fields:
Employee-ID, SIN, Name, Department, Address, Salary
Book Record:
ISBN, Title, Author, Publisher, Address, Price
Minimum functionality of your program:

- Must be well documented - internally and externally
- should be able to deal with at least one of the above record types
- Display the records in sorted order based on either of the two keys.
- Add a record
- Search for a record with a given key value
- Update the value of a field given the unique key value for the record

2. This part is worth 50 Points Write a program(s) which uses bubble sort, selection sort, insertion sort, merge sort and quick-sort algorithms to sort a data set in ascending order. Use some optimization techniques in the algorithms: for example, when the size of the list to be sorted is less than about 10, use the insertion sort algorithm. For data, use a random number generator to generate integer ’key values’ of sufficient number of decimal digits.
Make sure to count and display the total number of comparisons and moves/exchanges and show these values for 100, 1,000, 10,000, 100,000 and 1,000,000 elements to be sorted. Tabulate the results. Include the details of the hardware of the system used to make these evaluations.

Upload a single file to the course manager before end of the deadline, Suggested name of the file GnnA3.zip or GnnA3.bgz or GnnA3.rar depending on the compression scheme used (nn is your group number!) The programs should have all the internal and external documentations, and include all source code and the data used and the result produced. The marker has the option of testing the code submitted and using the same or similar data set.