CS-263: Design and Analysis of Algorithms Laboratory

LAB Assignment II

Course Instructor: Dr. Dibyendu Roy Due: Oct 27, 2023, 11:59 pm

Instructions: Code must be written in C language and it must be well commented. Write name and roll number on the top of your code. Submission of code in any other file extension (.pdf, .docx etc) will not be accepted. The file name of your code will be YOUR ROLL-NO.c

Let $S = \{0, 1, \ldots, 19\}$ be a set. Let a, b be two arrays of size 4 selected from S. We will define an ordering between a, b denoted by $<_o$. Here $\big[a[0], a[1], a[2], a[3]\big] <_o \big[b[0], b[1], b[2], b[3]\big]$ if either $(a[0]+a[1]+a[2]+a[3]) < (b[0]+b[1]+b[2]+b[3])$ or $(a[0]+a[1]+a[2]+a[3]) = (b[0]+b[1]+b[2]+b[3])$ and $a[i] > b[i]$ for the largest i for which $a[i] \neq b[i]$. According to this ordering $[0,0,2] <_o [0,1,1] <_o [1,0,1] <_o [0,2,0] <_o [1,1,0] <_o [2,0,0]$. Implement the `insert()`, `pop()`, `build heap()`, and `heap sort()` heap operations on arrays of size 4 selected from S. Your program should have the following features.

1. Allow the user to input a list of arrays (one by one) of size 4 (each element is separated by comma) from S, and use the `build heap()` operation to create a min heap with the given arrays. Display the heap to the user. Where vertex indices are also printed.

2. Allow the user to insert a single array into the heap created in Option 1. Display the resulting heap.

3. Allow the user to pop an array from the heap created in Option 1. Display the resulting heap.

4. Display the elements in the heap in sorted order by using the `heap sort()` operation.

5. Allow the user to provide n, k, r so that the computer may generate k many random arrays of size r from $S = \{0, 1, \ldots, n-1\}$. The program then sorts the arrays with `heap sort()`. Please allow your program to run continuously until the user selects the "Quit" option.