Haussmann, Benjamin

Diaz – Winter 2018

CS 241-01

Section 1.

Build a Max Heap using an array implementation. Generate random or sequential integers to fill the max heap and then print out the number of swaps, average number of swaps, and, if applicable, an optimal input for minimal swaps.

Section 2.

The program is designed to allow the user to choose from two options. The first option allows the user to generate 20 sets of 100 randomly generated integers to fill 20 max heaps. It will then calculate the average number of swaps across those 20 max heaps. The second option will allow the user to fill a max heap with 100 sequential integers from 1 to 100. It will then display the number of swaps necessary and output the first 10 integers of the heap. Following that, it will perform 10 removals and output the first 10 integers again.

Section 3.

To test the program, I ran it with option 2 selected many times to ensure that it would give back the same result each time as it was taking in the same input each time. For option 1, I first tested with some hard-coded inputs that I had already solved and ensured accuracy. Then to test the random number generator, I ran option several times and ensured that all numbers were similar to each other. As each result is the average of 20 runs, they should all be relatively similar, and they were. I received an average of 107 swaps for the insertion method and 68 swaps for the optimal method. I also checked and handled errors related to incorrect input. Whether the incorrect input is a wrong integer selection or a user input letter or special character, the program will identify the input as invalid and request a proper input.

Section 4.

I had a hard time with the checkCapactiy() method. Unfortunately, it took a while to realize that I was never copying over the old array when I doubled the size and created a new one. Once that problem was fixed, I stopped getting the NullPointerException’s. Another issue I had while writing this program was due the error handling of an invalid user input. I was able to catch invalid integer entries fairly easily, but I ran into issues handling the InputMismatchException’s I would get when anything other than a number was inserted. When I would catch the exception, the program would begin generating an endless loop of printing out “Invalid Selection. Please try again: “. I was user the nextInt() function of scanner and as such, the end of the line was never reached when a character was inserted. I solved the issue by adding a nextLine() to that catch.