CSCI181AF: Advanced Data Structures
09/014/2022

Writeup HW #2

Introduction

In this experiment, we utilized C++ code to compare two different algorithms for "heapifying" a randomized C++ vector into a max-heap. In order to produce the heap from the vector, we utilized two different algorithms, heapify_topdown and heapify_bottomup. The purpose of this writeup is to determine if there is a statistically significant difference in the performance of the two algorithms by comparing the average number of swaps in heapifying $n = 10, 10000, 10000000$ length vectors of values $[0, n-1]$ and the resulting 95% confidence interval generated to determine which algorithm used fewer swaps.

Data

heapify_topdown works by taking a vector, iterating from the beginning of the vector, and calling the function, bubbleUp on each element in the vector which "bubbles up" (swaps upward) the child values until the inequality $child \leq parent$ is met by focusing on the child element. heapify_bottomup iterates through a vector calling the function sinkDown on each element which swaps the parent and child nodes such that the parent can "sink" to their proper place in the heap.

Number of Points	Average	Standard Deviation	95% Confidence Interval
10	5.8	2.3	(5.00, 6.66)
10000	12753.7	136	(12705.03, 12802.37)
10000000	12815425.83	5305	(12813527.55, 12817324.12)

Table 1: Top Down Heapify Results

Number of Points	Average	Standard Deviation	95% Confidence Interval
10	4.57	1.33	(4.09, 5.04)
10000	7428.10	42.23	(7412.99 ,7443.2)
10000000	7440467.33	1714.24	(7439853.90, 7441080.765)

Table 2: Bottom Up Heapify Results

Analysis

Using Tables 1 and 2 to compare the performances of our two algorithms, it is noticeable that $n = 10$, there does not seem to be a statistically significant difference number of swaps, as indicated by the fact that the confidence intervals for each algorithm overlap. However, it is clear that as the vectors scale, as in our length 10000 and 10000000 vectors, there is a difference between the two algorithms as shown in the confidence intervals having no overlap and the Bottom Up algorithm having fewer swaps on average. Therefore, we can conclude that there is a statistically significant difference in the performance of the two algorithms in favor fewer swaps in the heapify_bottomup algorithm for larger vectors, as indicated by our $n = 10000, 10000000$ cases.