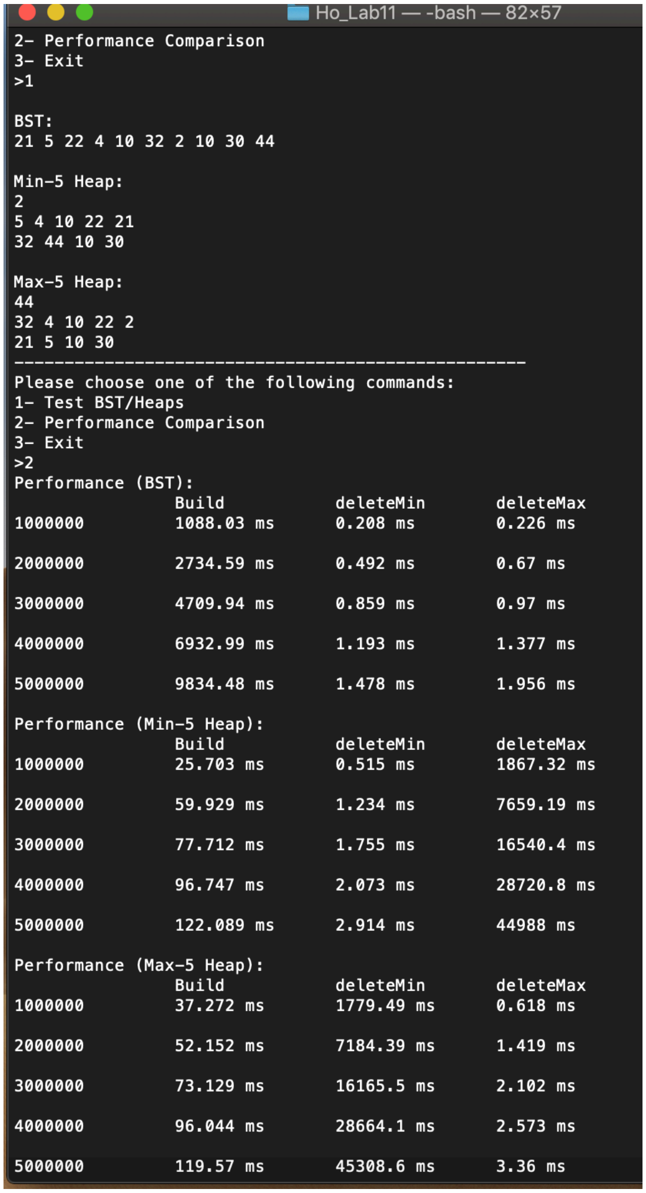
Lab11 Report
Tsung Yu Ho
2911531
Pandya Dhwani

Introduction:

We need record “Build” “Deletemin” and Deletemax” time for BST, Min-5 Heap and Max-5 Heap. The first process is to Insert 1 millions random number into Heap/BST, the range of the random number should be 1 to 5 millions, then we deletemin 0.001 m and deletemax 0.001m . We did it five times then record the average time for each for them. After these we repeat these steps with random number of size 2m 3m 4m 5m.

CPU time recording in C++



```
2- Performance Comparison
3- Exit
>1

BST:
21 5 22 4 10 32 2 10 30 44

Min-5 Heap:
2
5 4 10 22 21
32 44 10 30

Max-5 Heap:
44
32 4 10 22 2
21 5 10 30
---------------------------------------------------
Please choose one of the following commands:
1- Test BST/Heaps
2- Performance Comparison
3- Exit
>2
Performance (BST):
                Build           deleteMin       deleteMax
1000000         1088.03 ms      0.208 ms        0.226 ms

2000000         2734.59 ms      0.492 ms        0.67 ms

3000000         4709.94 ms      0.859 ms        0.97 ms

4000000         6932.99 ms      1.193 ms        1.377 ms

5000000         9834.48 ms      1.478 ms        1.956 ms

Performance (Min-5 Heap):
                Build           deleteMin       deleteMax
1000000         25.703 ms       0.515 ms        1867.32 ms

2000000         59.929 ms       1.234 ms        7659.19 ms

3000000         77.712 ms       1.755 ms        16540.4 ms

4000000         96.747 ms       2.073 ms        28720.8 ms

5000000         122.089 ms      2.914 ms        44988 ms

Performance (Max-5 Heap):
                Build           deleteMin       deleteMax
1000000         37.272 ms       1779.49 ms      0.618 ms

2000000         52.152 ms       7184.39 ms      1.419 ms

3000000         73.129 ms       16165.5 ms      2.102 ms

4000000         96.044 ms       28664.1 ms      2.573 ms

5000000         119.57 ms       45308.6 ms      3.36 ms
```

Performance Comparison:

Overall Comparison:

To build the Min-5 heap and Max-5 heap is faster than BST in my results. I think it's making sense min-5 heap and max-5 heap is using bottom-up approach so we insert the number first then heapify it, the big O of build this two are O(n). The build in BST takes way longer than in my results. Because the complexity is O(nlgn), it needs to be sorted then insert.

In Build Performance: Min-5 heap ≈ Max-5 heap >BST(Same as my reults)

To delete min in Min-5 heap and Max-5 heap is different. To delete min in Max-5 heap should be O(n) because we need to go through the whole heap to find the min value. But delete max should be O($\log_5 n$). To delete min in BST, which its' worst case is also O(n) since we might need to go all the way to the left to find the min value. However, we are discussing the worst case complexity but we are inserting millions of numbers, it's hard for us to have so such a tree that we need to traverse millions time like size to find the min value. So I think it makes sense that delete min is faster in BST than Max Heaps. However, BST delete min is also faster than Min-5 Heap, I think it's because of the heapify process.

In Deletemin Expected Performance: Min-5 heap > BST >Max-5 heap
My Result : BST > Min-5 heap> Max-5 heap

To delete max in Min-5 heap and Max-5 heap is still in opposite way. To delete max in Min-5 heap should be O(n) because we need to go through the whole heap to find the max value. But delete max in Max-5 heap should just be O($\log_5 n$). To delete max in BST, which it's worst case is also O(n) since we need to go all the way to the right to find the max value. As I stated before, we are discussing the worst case complexity. However usually it didn't take that long the find the max value in BST because it's hard to have tree that need us to traverse right millions time like size to find the max value. Hence, I think it makes sense that my deletemax is faster in BST than Minheaps. However, BST delete max is also faster than Max-5 Heap, I think it's because of the heapify process.

In Deletemax Expected Performance: Max-5 heap> BST > Min-5 heap
My Result: BST > Max-5 heap >Min-5 heap

Conclusion:

In this lab, my BST is working well to print out the time, but Min-5 Heap and Max-5 Heap will have core-dumped when I it gets to insert 3 million numbers. However, my testing is printing out he correct output for both the Heap. So it takes some time for me to find out why. Then I realized I forgot to add statements that if index is bigger than 10millions its shouldn't be checked, because checking child will need to do (5*index+1). It will be out of range if I check the not-existing index and cause core dumped. My deletemin in Min5-heap and delete max in Max5-heap has around the same output time. My deletemin in Max5-heap and deletemax in Min5-heap also have around the same output time due to the same logic. For the deletemin and deletemax is slowers than BST is probably because we need to get the number from last element and then heapify.