Lab report

James Ballard

1. The overall organization of your experiment

The purpose of this lab was to implement a min 3-heap and compare its functionality with that of a binary search tree. We did this with two steps taking the average of the time to build the structure for a given amount of values and then testing the time to build the function.

1. Data generation

To generate the data we did something very similar to our last lab. Using different srand keys we generate random arrays of size n upon which we added to the structures. Then using the same keys we randomly decided to do either an insert or remove of some kind (min, max, or a particular) on the structures

1. Summary of the Results

Well the min heap was clearly better at building. It was quite bad at other functions mainly because (removing a particular value and removing the max are very inefficient) in a min heap

1. Conclusion

Well a min heap can be constructed far quicker than a BST, and function’s such as insert and delete min operate very quickly functions such as delete(x) and delete max are looking if down the values of an array making them really bad. So if you don’t have to delete random values in the segment or the maximum value ever the min heap would be a better choice then binary search tree. But if you have to delete certain values or the max ( although you could make a max very easily k-heap) then a Binary Search might be the better option.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | N | 1 | 2 | 3 | 4 | 5 | average |
| min heap | 50000 | 0.003502 | 0.002572 | 0.002569 | 0.002586 | 0.002562 | 0.002807 |
|  | 100000 | 0.006956 | 0.004956 | 0.005207 | 0.004998 | 0.004995 | 0.005529 |
|  | 200000 | 0.015023 | 0.009869 | 0.009778 | 0.012546 | 0.012555 | 0.011804 |
|  | 400000 | 0.026535 | 0.019836 | 0.02529 | 0.026347 | 0.026119 | 0.024502 |
|  |  |  |  |  |  |  |  |
| functions | 50000 | 0.561335 | 0.560493 | 0.574052 | 0.570269 | 0.555176 | 0.566537 |
|  | 100000 | 2.26977 | 2.28541 | 2.27388 | 2.26722 | 2.26711 | 2.27407 |
|  | 200000 | 8.97326 | 8.95724 | 9.02193 | 9.06191 | 8.95924 | 9.003585 |
|  | 400000 | 36.1351 | 36.404 | 36.2585 | 36.7385 | 36.2475 | 36.38403 |
|  |  |  |  |  |  |  |  |
| BST | 50000 | 0.039918 | 0.035701 | 0.045431 | 0.055419 | 0.062438 | 0.047781 |
|  | 100000 | 0.076253 | 0.082286 | 0.124862 | 0.174936 | 0.194564 | 0.13058 |
|  | 200000 | 0.174809 | 0.220045 | 0.37026 | 0.444835 | 0.478627 | 0.337715 |
|  | 400000 | 0.436175 | 0.645307 | 0.936503 | 1.0702 | 1.12371 | 0.842379 |
| Begin of functions | 50000 | 0.003749 | 0.004195 | 0.006085 | 0.007018 | 0.007679 | 0.005745 |
|  | 100000 | 0.008414 | 0.009682 | 0.016727 | 0.016538 | 0.018096 | 0.013891 |
|  | 200000 | 0.019438 | 0.02221 | 0.034963 | 0.042229 | 0.045554 | 0.032879 |
|  | 400000 | 0.048028 | 0.056222 | 0.091358 | 0.09913 | 0.108097 | 0.080567 |

Lab report