Skip Lists

Vincent Chen
CS 610102: Data Structure and Algorithm

Final Project
Spring 2019

1. Introduction

I use srand(time(NULL)) to change different random seed by time and use function rand() % 2 == 1 in insert function to have 50% chance grow up to next level.

In test and auto test, I create two vectors and shuffle them randomly as an input data. For example, if I want to do n times insert, delete, find_success and closest after function, the odd_array will be [1, 3, ……, 2n-1] with random shuffle every times. if I want to do n times find_fail function, the even_array will be [0, 2, ……, 2(n-1)] with random shuffle every times.

2. Average time

I use auto test function to run all five operations 100 times and take the average by the iteration of 128, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072.

Below is the chart of total time for each operation, the unit is micro second:

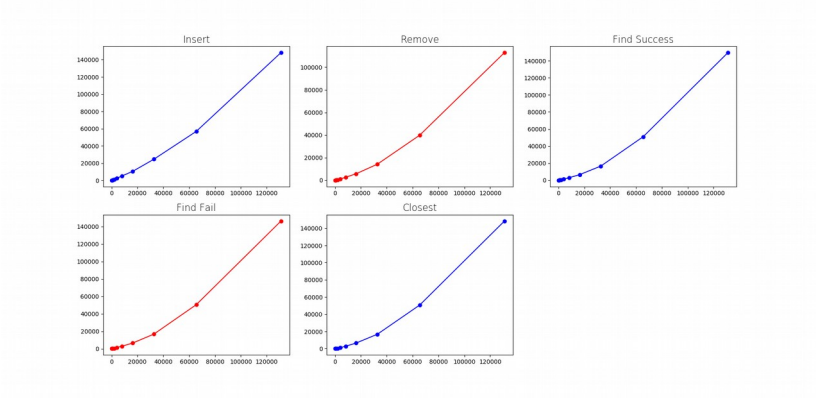
	Insert	Remove	Find Success	Find Fail	Closest
128	136.63	53.24	51.08	54.08	50.86
512	244.35	103.55	102.05	104.45	100.36
1024	523.88	222.35	218.41	232.5	216.54
2048	918.82	419.24	428.08	431.81	431.59
4096	2346.28	1101.94	1209	1281.72	1204.05
8192	5110	2622.85	2943.24	2816.21	2762.83
16384	10406.4	5775.64	6552.35	6472.51	6575.58
32768	24465.4	14218.1	16596.9	16718.5	16774.7
65536	56753.9	39910.4	50994	50487.2	50736.7
131072	148134	112961	149494	146514	148384

Below is the chart of unit time for each operation, the unit is micro second:

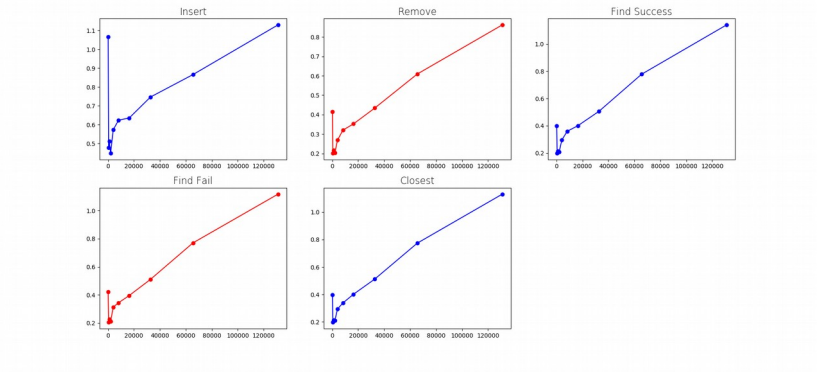
	Insert	Remove	Find Success	Find Fail	Closest
128	1.067422	0.415938	0.3990625	0.4225	0.397344
512	0.477246	0.202246	0.19931640625	0.204004	0.196016
1024	0.511602	0.217139	0.21329101563	0.227051	0.211465
2048	0.448643	0.204707	0.2090234375	0.210845	0.210737
4096	0.572822	0.269028	0.29516601563	0.31292	0.293958
8192	0.623779	0.320172	0.35928222656	0.343776	0.33726
16384	0.635156	0.352517	0.39992370605	0.395051	0.401342
32768	0.746625	0.433902	0.50649719238	0.510208	0.511923
65536	0.865996	0.608984	0.77810668945	0.770374	0.774181
131072	1.130173	0.861824	1.14054870605	1.117813	1.13208

3. Graph

Below is the graph of total time for each operation, the unit is micro second:



Below is the graph of unit time for each operation, the unit is micro second:



4. Conclusion

According to the graph of unit time, the time complexity is between O(n) and $O(\log_2 n)$.

The 128 iteration have huge deviation, I thing it is because the iteration is too small. After 512 iteration, the time looks reasonable.

There are basically no difference in spending time of find_success, find_fail and closest after operation. The insert operation has the most spending time and delete operation has the least.