Claire Jensen

Data Structures

Professor Gardner Spring 2021

Homework 4

What did your results show? That is, how does the runtime of find compare, as n grows, for the low, moderate, and high values of p?

The runtime increases as p gets lower. When p is set to something very low, such as .001, the runtime is very slow as the program runs through more numbers. However, when set to something higher, such as .95, the runtime is faster. When p is set to .5, it runs about in the middle of the two, as a reasonable runtime. The best value of p is 0.5 to make an even list.

Why do you think the results looked the way they did? What does a skip list look like when p is very low? What about when p is very high? How do those list structures lead to the results that you saw?

I think they looked this way because when p is very small, the runtime went up exponentially as the operations went on. This is because when the value of p is low, the list will not have a very high height, if higher than 0 because there's a very low chance that any nodes will be increased higher than 0. The same thing happens when p is very high- the nodes will all have a very similar height but will be very tall. Both situations make the SkipList very hard to skip over because the levels are all the same height, ultimately defeating the purpose of skipping over nodes in the list. Any value of p that is not exactly .5 will skew the heights of the nodes in a certain direction and make it difficult to efficiently skip nodes this way.

Did you observe anything else interesting in your SkipList experiments?

I think it was interesting that the first few numbers took milliseconds to compute, but as the experiment went on, the runtime increased by a lot and took about 25-30 seconds each instead of milliseconds.