Lecture We will talk about amortised analysis and see a another kind of binary search trees, called splay trees. You should read about

- amortised analysis in either Lecture 15 in the notes by Jeff Ericsson or CLRS chapter 17.
- Splay trees in section 16.5-16.6 in the notes by Jeff Ericsson.

The notes by Jeff Ericsson can be found here: http://www.cs.illinois.edu/∼jeffe/teaching/algorithms (also available on CampusNet). The Chapter from CLRS is also available on CampusNet.

Exercises

An exercise labeled with * means that the exercise is difficult. You can skip the exercise and get back to it later if you have time.

1 Splay trees

1.1 Show the splay tree that results from inserting the keys 41, 38, 31, 12, 19 , 8 in this order into an initially empty tree.

1.2 Show how the tree looks after deleting 31.

2 Aggregate and accounting Suppose we have a data structure where the cost $T(i)$ of the ith operation is:

$$T(i) = \begin{cases} 2i & \text{if } i \text{ is a power of 2,} \\ 1 & \text{otherwise.} \end{cases}$$

What is the amortized running time of the operation? Use both the aggregate method and the accounting method to analyze the amortized running time.

3 Potential functions: Doubling arrays In this exercise we consider dynamic tables with insertions only (no deletions) using the doubling technique. Can we show that the amortized cost of an insertion is $O(1)$ using the following potential function: $\Phi(D_i) = k$, where k is the number of elements in the array?

4 Set Union In the Set Union problem we have n elements, that each are initially in n singleton sets, and we want to support the following operations:

- Union(A,B): Merge the two sets A and B into one new set $C = A \cup B$ destroying the old sets.
- SameSet(x,y): Return *true*, if x and y are in the same set, and *false* otherwise.

We can implement it the following way. Initially, give each set a color. When merging two sets, recolor the smallest one with the color of the larger one (break ties arbitrarily). To answer SameSet queries, check if the two elements have the same color.

4.1 Analyze the worst case cost of the two operations.

4.2 Show that the amortized cost is $O(\log n)$ for Union and $O(1)$ for SameSet. That is, show that a any sequence of m unions and l SameSet operations takes time $O(m \log n + l)$.

Hint: Give a bound on the number of times an element can be recolored.

5 Play trees Professor Bille suggests a simpler version of Splay Trees that he calls Play trees. Play Trees are similar to Splay Trees but uses only single rotations when splaying.

5.1 What is the amortized cost of splay(x) if we only use single rotations? Analyze it with the same potential function as we used for Splay Trees.

5.2 What is the total (actual) cost of first n inserting elements with keys $1, 2, 3, \ldots, n$ in that order in a Play Tree and then searching for $1, 2, 3, \ldots, n$ (in that order)?

5.3 Professor Bille claims that the amortized cost of the operations insert, search and delete in Play trees is $O(\log n)$. "You just need to use a more sophisticated potential function" he says. Could he be correct?

6 Dynamic hashtable Explain how to make a dynamic hashtable with insertions using the doubling technique. Your solution should use $\Theta(n)$ space, where n is the number of elements in the hashtable. What is the insertion time of your solution?

7 Deamortization of dynamic tables It is sometimes possible to *deamortize* data structures, i.e., getting the same bounds worst case as amortized by doing the work in the "background". Show that you can get worst case constant insertion time in dynamic tables, while still having space usage $O(n)$, where n is the number of elements currently in the table.

Hint. Use the doubling technique, but spread out the work on all the insertions.

Puzzle of the week: Princesses You are a young Prince from the country Algo. The King in the neighboring country Logic has 3 daughters. The oldest one always tells the truth, the youngest one always lies, and the middle one sometimes lies, sometimes tells the truth.

You want to marry either the oldest one or the youngest one (since you know she *always* lies that is as good as the one always telling the truth). The only one you don't want to marry is the middle one.

The king is a sneaky man, and he tells you, that you can ask *one* of the daughters *one* question. The question should be one with a yes/no answer. After that you have to choose which one to marry. They all look alike, so it is not possible for you to determine which one is which by looking at them.

What question should you ask, and which one should you then pick?

Implementation of dynamic tables [CJ] Implement your own dynamic table for integers (without using the built-in versions). Your dynamic table must support insertion, deletion, printing of inserted elements and reporting of the table size.