Reading

See webpage.

Exercises

1 [w] **MultiPush** Consider the stack with MULTIPOP and add the following additional operation:

- MULTIPUSH(x, k): add k duplicates of x onto the stack S.

Determine how this affects the amortized running time of (all of) the stack operations. Is the amortized running time still constant?

2 [w] **Queues** Show how to implement a queue using two stacks (and no other data structures) such that both the ENQUEUE and DEQUEUE operations take amortized constant time.

3 **Set Union** The *set union problem* is to maintain a dynamic family of sets supporting the following operations.

- INIT(n): Create n singleton sets.
- UNION(A, B): Create the union $C = A \cup B$ and return a pointer to C. The sets A and B are destroyed.
- SAMESET(x, y): Return true, if x and y are in the same set, and false otherwise.

Note that the problem is closely related to the union find problem (how exactly?). Consider the following solution idea. For INIT(n) we create the n singleton sets and assign each element a distinct *color*. For UNION(A, B) we re-color the elements in the smaller set with the color of the larger set (breaking ties arbitrarily). Finally, for SAMESET(x, y), we compare the colors of x and y and return true if and only if they match.

Solve the following exercises.

3.1 Do a standard analysis of the running time of the data structure for all of the operations.

3.2 Do an amortized analysis of the data structure for the relevant operations. *Hint:* Give a bound on the number of times an element can be recolored.

4 **Binary Heaps** Consider a binary (max-)heap with n elements.

4.1 [w] Recall the basic structure of a heap and how the INSERT and EXTRACT-MAX operations work. Recall the standard analysis for the running time of INSERT and EXTRACT-MAX.

4.2 Do an amortized analysis to show that the amortized running time of INSERT is $O(\log n)$ and the amortized running time of EXTRACT-MAX is constant. Use the potential method. *Hint:* Use the depth of the nodes as part of your potential function.

5 **Amortized Analysis Methods** Consider a data structure where the running time T_i of the ith operation is:

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

Solve the following exercises.

5.1 Analyze the amortized running time of the operation using the aggregate method.

5.2 Analyze the amortized running time of the operation using the accounting method.

5.3 [$*$] Analyze the amortized running time of the operation using the potential method.

6 Dynamic Ordered Sets Consider the dynamic binary search data structure. Assume that we can support SEARCH in $O(\log n)$ time. Solve the following exercises.

6.1 In the lecture, we ignored how to handle *duplicates* during INSERT operations, i.e., inserting the same value multiple times. Explain why this is an issue and give an efficient algorithm to handle duplicates.

6.2 [$*$] Show how to implement the DELETE operation in $O(\log n)$ amortized running time.

7 Billy the Rabbit (Exam 2017) The rabbit Billy lives with his family in a rabbit hole. The rabbit Billy is very shy, and he loves carrots. He knows that there are carrots somewhere on the path going from the rabbit hole into the woods. But he does not know how far away the carrots are.

Billy is afraid to leave home alone, so he comes up with the following strategy. In the first round, he takes 1 step and then goes back to the hole. In the next round, he takes 2 steps and then goes back; in the third round, he takes 4 steps, and so on. That is, in each round he takes twice as many steps as in the previous round. Assume the carrots are n steps away. Solve the following exercises.

7.1 How many rounds does it take before Billy finds the carrots?

7.2 What is the total number of steps Billy takes before he finds the carrots? (An asymptotic answer is satisfactory.) Explain your answer.

7.3 What is the amortized number of steps that Billy takes in a round? (An asymptotic answer is satisfactory.) Explain your answer.

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?