Lecture We will talk about amortised analysis. You should read CLRS chapter 17 (available on DTU Learn.).

Exercises

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

1 MultiPush Consider the stack from the lectures with the following additional operation: k-Push(k,e) which pushes the element e onto the stack k times. Is the amortized cost per stack operation still $O(1)$?

2 Analysis 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 Queues Show how to implement a queue with two stacks such that both Enqueue and Dequeue takes amortized time $O(1)$.

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 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 Binary Heaps Consider a binary heap of size n (the root stores the smallest element).

5.1 Recall the worst case cost of Insert and Extract-Min, and how they work.

5.2 Use the potential method to show that the amortized cost of Insert is $O(\log n)$ and the amortized cost of Extract-Min is $O(1)$.

Hint: Use the depth of the nodes as part of your potential function.

6 Billy the Rabbit (Exam 2017) The rabbit Billy lives with his family in a rabbit hole.

Amortized 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 very shy and 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, on the third round he takes 4 steps, etc. That is, in each round he takes twice as many steps as in the previous round. Assume the carrots are n steps away.

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

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

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

Randomized Billy is both shy and very forgetful. Last week he hid k carrots in k different bushes. Now he is hungry, but he forgot in which of the b bushes around the rabbit hole he hid the carrots. To find a carrot he now does the following. In each round he goes to a random bush and checks if there is a carrot. If not he runs back to the rabbit hole and tries again. Since he is very forgetful he might go to the same bush again in the next round.

6.1 What is the expected number of bushes that Billy visits before he finds the first carrot? Explain your answer.

6.2 After eating the first carrot Billy is still hungry, so he keeps looking for two more carrots. He does not remember where he already has found carrots, so he might go to these bushes again. What is the expected number of bushes that Billy visits before he has found 3 carrots? Explain your answer.

7 [†] **Implementation of dynamic tables** 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.

8 **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?

9 [$**$] **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?