

ADS 2023 spring Week 9 Exercises

Exercises for Algorithms and Data Structures at ITU. The exercises are from *Algorithms, 4th Edition* by Robert Sedgwick and Kevin Wayne unless otherwise specified. Color-coding of difficulty level and alterations to the exercises (if any) are made by the teachers of the ADS course at ITU.

From Thore's notes - Green

- a) Bob is a runner. He has been running 4 km every day of the week, except in the week-ends. What is the worst case number of kilometers he runs per day? He started this new workout schedule on a Saturday. What is the amortised number of kilometers he runs per day?
- b) Ran is also a runner. He rolls a die every morning and runs as many kilometers. (i) What is the expected number of kilometers Ran runs per day, assuming he has a perfect die? (Or what about enchanted dice that always come up 1? Or cursed dice that always come up 6?) (ii) What is the worst-case number of kilometers Ran runs per day? (iii) For any sequence of days starting on a Saturday, what is the worst-case amortised number of kilometers Ran runs per day? (Note that this answer must hold *even if his dice are cursed.*)
- c) My phone company charges 10 DKK for 1 minute of voice call. This is exorbitant, but I never use my phone for making a phone call anyway. According to the contract, I have to pay at least 100 DKK per month for voice calls whether I use them or not, but the contract automatically 'rolls over' the unused minutes to the next month. I have to call my mum for Christmas for a 2-hour call. Describe the expense in terms of 'money I spend on voice calls each month' in the worst case and in the amortised sense.
- d) A multiride ticket in the Danish amusement park Tivoli costs 200 DKK and is valid for 10 rides. What is the worst case cost for a single ride? What is the amortised cost for a ride? (*Careful!*)

Old exam set 110530: 2 - Green

See question 2 in the exam set bads-110530.pdf on learnit.

1.4.29a - Yellow

This was a red exercise in week 3. The focus this week is on the argument for the amortized number of stack operations.

Steque with two stacks. Design a steque (a stack-ended queue or steque is a data type that supports **push**, **pop**, and **enqueue**) with two stacks so that each steque operation takes a constant amortized number of stack operations.

Old exam set 120531: 2 - Yellow

See question 2 in the exam set bads-120531.pdf on learnit. Note: g) and h) are also covered in the quiz.

Set Union - Red

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.

- a) Analyze the worst case cost of the two operations.
- b) 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 s SameSet operations takes time $O(m \log n + s)$. Hint: Give a bound on the number of times an element can be recolored.

1.4.31a - Red This was a red exercise in week 3. The focus this week is on the argument for the amortized number of stack operations. Deque with three stacks. Design a deque (a double-ended queue that supports pushLeft, pushRight, popLeft and popRight) with three stacks so that each deque operation takes a constant amortized number of stack operations.