Weekplan: Range Minimum Queries

Philip Bille Inge Li Gørtz

Reading

Notes on Partial Sums and Dynamic Arrays, Chapter 1, 1.1, and 1.3.

Exercises

1 $[w]$ **Segment Trees** Consider the array $A = [2, 7, 1, 8, 0, 6, 3, 6]$. Draw the segment tree for A and write down the array corresponding to the segment tree. Show how the tree and the array look after the following operations: ADD(2.5) and ADD$(6, -2)$.

2 $[w]$ **More Segment Trees** Which of the following arrays is a correct segment tree:

$$A_1 = [-, 0, 3, 0, 3, 6, 0, 2, 8, 3, 9, 6, 7, 0, 2, 5]$$

$$A_2 = [-, 1, 1, 4, 1, 3, 4, 6, 1, 2, 3, 4, 5, 6, 7, 8]$$

$$A_3 = [-, 1, 1, 1, 1, 1, 5, 7, 1, 4, 7, 4, 1, 5, 7, 7]$$

3 Range Sum Queries Solve the exercise "sums" from the weekplan on priority queues and heaps (no longer any $[*]$ or $[**]$ on the subexercises).

4 Pizza Prices Bob lives on P Street in Algotown. There are n buildings on the street, and each house has a pizzeria and an apartment. The pizza price in building k is p_k, and the prices often change.

4.1 Bob often visits his friends in the other buildings on the street, and when they meet, they always eat pizza. They are tired, so they don't want to walk too long. Help Bob by giving a data structure that supports the following operations:

- UPDATE(k, v): change the price p_k to v.
- FINDCHEAPPIZZA(k, w): return the cheapest pizzeria within w buildings of apartment k.

4.2 $[*]$ It's the end of the month, and Bob and his friends are low on cash and need to find a pizza that costs no more than p. They still want to walk as little as possible. Help Bob by giving a data structure that can support the following operations:

- UPDATE(k, v): change the price p_k to v.
- FINDNEARESTPIZZA(k, p): return the nearest pizzeria to apartment k with a price of at most p. If no such pizzeria exists, return "No money, no pizza!".

5 $[*]$ **Range Updates** In the *range update problem*, we want to preprocess an array A to support the following operations efficiently:

- ADD(i, j, k): Add k to each of the entries $A[i] \ldots A[j]$.
- LOOKUP(i): Return the value $A[i]$.

Give an efficient solution to solve the range update problem.