DAT600: Algorithm Theory

Set ADT and Dynamic Programming

Assignment – 2: Compulsory

Code of Conduct

The principle is trust, participation and collaboration for better learning.
So please, do your best to learn and don't try to cheat. It is allowed and encourage to collaborate with others but for the purpose of learning, not just copy-pasting. You are allowed to consult and get inspired by external resources, but you should mention them in your report. NB! To get approval you will have to explain your work to the course responsible during the lab sessions.

Delivery

Write a short and concise report in which you solve the following tasks. Add code sections when ever necessary, and a link to your GitHub repository if you find more convenient.

Task 1

a) Implement

Implement the following Set Abstract Data Type (ADT):

- `Element`: It contains the following fields: a key k and data (whatever data you like)
- `build(X)` in $O(n \lg n)$: Given a collection of elements X it creates a set.
- `find(k)` in $O(\lg n)$: Given an element key k,returns its key, data, and a bool. If the element does not exist, it returns nil, nill, false, otherwise it returns key, data, true.
- `insert(x)`in $O(\lg n)$: Inserts and element in the set if it does not exists
- `delete(k)`in $O(\lg n)$: Delete element with key k if it exists
- `find_min()`in $O(\lg n)$: Find element with minimum key
- `find_max()`in $O(\lg n)$: Find element with max key
- `find_prev(k)`in $O(\lg n)$: Find the next element (with key greater than k)
- `find_next(k)`in $O(\lg n)$: Find the previous element (with key less than k)

b) Explain

Justify your claims on the runtime complexity by analyzing the algorithm or by testing.

Task 2

a) Implement

Implement the disjoint SEt Abstract Data Type (ADT) wit the following operations:

- `MAKE-SET(x)` in $\Theta(1)$: Makes a set with x as its only element
- `FIND-SET(x)` in $O(\alpha(n))$ where $\alpha(n) << \lg n$: Find the set that x belongs to
- `UNION(x,y)`in $O(\alpha(n))$ where $\alpha(n) << \lg n$: Makes a union of the sets that x belongs to with the set that y belongs to.

b) Explain

Justify your claims on the runtime complexity by analyzing the algorithm or by testing.

Task 3 Dynamic and Greedy Algorithms

a)

Explain in few words the conditions that must be satisfied for dynamic programming to apply.

b)

Explain in few words three potential strategies to solve a dynamic programming problems.

c)

Choose a dynamic programming problem and do the following:

- Formulating its recursive relation
- Solve it programmatically in a naive way (recursion as it is) and record its execution time.
- Solve it programmatically using the two techniques that were explained in the lectures, explain shortly your approach and record the execution time.
- Try to solve it programmatically using a greedy approach, explain why or why not the greedy approach applies. Record the execution time.

d)

Summaries in a paragraph or two your understanding of dynamic programming and greedy algorithms.