Algorithmen & Datenstrukturen

Woche 13

Marius Tomek, Nicolas Wehrli, Tim Rieder

19. Dezember 2022

ETH Zürich

Inhalt

Kurze Kommentare zur letzten Serie

Floyd-Warshall & Johnson

Exam prep

Homework 12

Peergrading 12.2

Kurze Kommentare zur letzten Serie

Kommentare zu Serie 11

11.2)

a) It isn't enough to argue I_b is contained in I_a, because it could also be completely outside of it. But with the assumption $pre(a) < pre(b) < post(a)$ we have $post(b)$ must also be in-between the a interval.

11.3)

It is easier to first modify the graph and then run an unmodified Algo instead of modifying the Algo to run on the original graph.

a) Can be solved in easier ways than masters solution (no duplicate of vertices, only edges from $v_s - R$, $R - I$, $I - G$, $G - G$, $G - F$, $F - v_d$.

b) Masters solution is probably as easy as it gets. But there are still variants.

Multiplying edges/nodes by a constant doesn't change the runtime of Dijkstra.

Kommentare zu Serie 11

Mistakes

- all exercises) ALWAYS DOUBLE CHECK YOUR ALGO AGAINST EDGE CASES!
- b) You can't reuse (a) 4 times for each possibility for the first vertex. You can't reuse (a) 4! times for each possible permutation of *RIGF*, because the best path could be longer than 4 vertices.
- b) If we want to reuse (a), we need all permutations ($4! = 24$), then we need repeating identical nodes to be possible (*RIGGF*), then we need repeating non-identical nodes to be possible (*RIGIGF*).

Floyd-Warshall & Johnson

Exam prep

Pseudocode

- Don't write pseudocode if you don't have to (you can always add 2-3 lines to your text if you think you can't explain one part of it in words, but it costs too much time and you can make more mistakes).
- Don't use `for i in range()`, `s.charAt()`, `s.substring()`, `final/static/private` or similar.
- Do use `Input: ...`, `Global variables: ...`, `a[1..m][1..n] initialized to 0`, `function f() {...}`, `if ... then`, `if ... do`, `for i=1..n do`, `for x←n downto 1 do` or similar.
- Always with runtime and short reasoning (helps to correct it, so it helps you).
- You can reuse Algo from lecture if it works exactly the same. Otherwise you need to write down the pseudocode.

Exam prep

General

- Make clean/readable/understandable solutions. It favours you when grading.
- READ the exercises carefully! You can miss so many points when you rush this.
- Solve old exams from VIS.

A&D

- You can program all data structures and algorithms once on your own.
- Solve DP Problems (https://cses.fi/problemset, https://spoj.com/, https://codeforces.com/, https://leetcode.com/).

Algorithms you really should know

- Binary search
- Heapsort (or one of those 3)
- PrioQueue (uses Heapsort), a bit of Stack/Queue/Binary search tree
- as many DP examples as you can find
- D/BFS, one for MSTs, Dijkstra, Bellman-Ford, (Floyd-Warshall, Johnson)

> *Implement: the above. Understand: all.*
> *In theory questions, they can ask specifically about one Algo.*
> *In coding, you should be fine knowing only one Algo per problem.*

Homework 12

Peergrading 12.2