1A Lent Algorithms

Supervision 6: MST, DAGs, Flows

Hayk Saribekyan

February 28, 2019

When you are asked to describe an algorithm, also analyse its time and space complexity. Mention how you would implement the solution.

In this supervision, we will finish shortest-path algorithms on graphs and work on tree and DAG algorithms. This example sheet is a hybrid of lecturer's example sheet 6 and mine[1]. Some of the problems added by me are not directly useful for the exams or are out of the scope of the course. These are marked with *. These problems, however, show how some of the standard algorithms that you have learnt can be used. If you have time, you should work on them as they will develop your intuition.

I would rather see a failed attempt or ideas handed in, than a beautifully written correct solution of a problem that you found easy. I do not mind if you skip a problem and write "I know how to do it." So focus on the concepts that you have most trouble with rather than writing down solutions to problems that are easy for you.

1 Trees, DAGs

Some problems are from `https://www.cl.cam.ac.uk/teaching/1819/Algorithms/ex6.pdf`

1. (DW) In a connected undirected graph with edge weights ≥ 0, let $u - v$ be a minimum-weight edge. Show that $u - v$ belongs to a minimum spanning tree. Does it have to belong to *all* MSTs?

2. How would you find a *maximum* spanning tree of an edge-weighted graph?

3. Given an MST for an edge-weighted graph G and a new edge e. Describe how to find an MST of the new graph.

4. Describe an algorithm that finds all edges that are part of *some* MST. *Make it run in $O(|E| \log |V|)$ time.

   Hint: Find some MST T. What can you say about edges that are not in T?

5. (DW) Give pseudocode for an algorithm that takes as input an arbitrary directed graph G, and returns a boolean indicating whether or not G is a DAG.

6. Describe an algorithm that determines whether the topological ordering is unique.

7. * Describe an algorithm that finds the longest path in a DAG.

[1] Lecturer's sheets: `http://www.cl.cam.ac.uk/teaching/1819/Algorithms/materials.html`

2 Flows

Some of the problems are from the Lecturer's example sheet 6[2].

1. Solve Q1, Q4, Q5 from the sheet.
2. In Question 5, how would you find the minimal set of disruptions that will cut off the two stations?
3. You would like to assign n people into m activities in such a way that the largest activity group is as small as possible. For each person, you know exactly which activities they can be part of. Describe an algorithm to find the smallest possible size of the largest group.
4. * Find an example of a graph for which the Ford-Fulkerson algorithm never terminates.
5. * Consider an $n \times m$ matrix, where each element is either 0, 1 or ?. Describe an algorithm that replaces ? with either 0 or 1 such that the sum of elements of row i is a_i, and the sum of elements in column j is b_j. Your algorithm should determine if it is not possible to do so.

Hints

- *To tree/dag question 7:* Use DP.
- *To flows question 3:* First, find an algorithm that checks whether, for a given integer s, it is possible to make the assignment in such a way that the largest group has at most s people.
- *To flows question 4:* The important thing to notice in this example is that one needs non-rational weights on the edges (try to prove that termination will occur otherwise).
- *To flows question 5:* You need to build some graph and convert the problem into a flow.

[2] `http://www.cl.cam.ac.uk/teaching/1718/Algorithms/ex6.pdf`