Second Round Theoretical Tasks

Swiss Olympiad in Informatics

March 4, 2017

Instructions

The solutions will be graded according to similar criteria as the first theoretical round. The most important criteria are correctness and asymptotic run time. The quality of the description and the arguments asserting the correctness will also be taken into account.

To describe an algorithm, you should structure your solution as according to the following guideline:

1. Describe the idea for an algorithm that solves the problem. The description should be understandable without looking at its source code.
2. Argue about the correctness of the approach.
3. Indicate asymptotic running time and memory usage.
4. Write an implementation in pseudo code. This part should contain the source code of the most important parts of the algorithm in something that ressembles some programming language. You can skip simple parts like input, output and can use mathematical expressions.

If some part of your solution can be used for multiple subtasks, it suffices to write him down only once and refer to it from other parts.

Fleptons

Mouse Stofl is a brilliant theoretical physicist and has just informed you about his newest discovery, the fleptons! A flepton pair spontaneously spawns at an arbitrary location in space and these two fleptons will then immediately separate. After the separation, these two fleptons will take some path through space until they collide into each other and disappear.

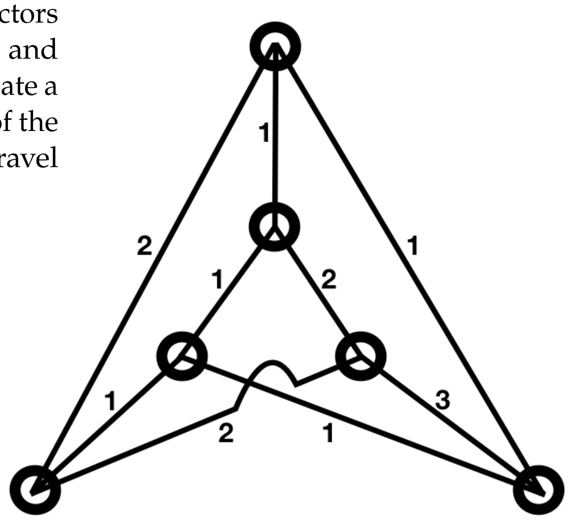
Stofl has shown the following curious property of fleptons: A flepton will never occupy a location in space twice, nor will it occupy a location in space which its flepton counter-part has occupied previously. A flepton only visits a location in space if it couldn't have visited it earlier (this means its entire path is a shortest path in space).

Your job is to help mouse Stofl receive the Nobel prize in physics, which he can only acheive by experimentally confirming the existence of fleptons. Mouse Stofl has already built a network of m molecular tubes and n detectors, in which he hopes to measure the flepton pair. A molecular tube is always between two detectors and, due to practical circumstances, only one molecular tube can possibly connect two nodes. A flepton always travels at the speed of light and therefore travels with constant speed. Since the tubes are so small, the fleptons are forced to travel in one direction (and can't travel back). The length of the tubes is measured by the time it takes for a flepton to cover its distance (in ns). You now have to help Stofl find two detectors between which fleptons might arise/decompose, so that Stofl can closely monitor these two detectors.

Formal description You are given a weighted undirected simple graph, find two *shortest* paths, which both start at the same vertex, both end at the same vertex, cover the same distance (edge weights added together) and do not share any other vertices.

Subtask 1: Solve the Example (10 Points)

Find said two shortest paths between two detectors in the following system of molecular tubes and mark them on the paper. The circles each indicate a detector and the numbers indicate the length of the tubes / the time it takes for the fleptons to travel through them in ns.



Subtask 2: Develop an Algorithm (50 Points)

Develop an algorithm for Stofl, that finds two possible paths for the flepton pair to move through, given any system of molecular tubes for which such paths exist. The fleptons have to start/end at the same detectors. Argue why your program is *correct*, write down the algorithm in *pseudocde* and analyse the *running time* and *memory usage* of your algorithm.

Subtask 3: Configuration for Detecting more Fleptons (40 Points)

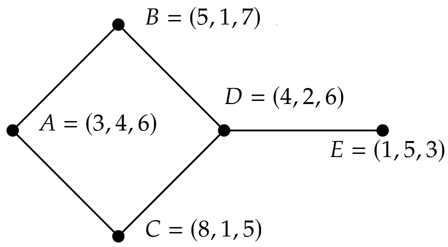
Yikes! Stofl has almost spent all his money on detectors and he still has not found any fleptons. Due to the tight budget, he can now only buy tubes of the same size. Additionally he wants to increase the probability of the detectors finding a flepton. Try to help him construct an entirely new connected configuration consisting of the new tubes with n detectors, such that every non-adjacent pair of detectors can form and destroy fleptons. You will receive more points if you need fewer tubes, for example a solution with $m = 2 \cdot n$ tubes will receive more points than a solution with $m = 5 \cdot n$ tubes.

Teleport

The mice on a distant planet travel between cities using a network of teleports. A teleport consists of three transmitters each operating on a different frequency. No two teleports have all frequencies equal.

A mouse can only travel between two teleports if they do not share a transmitter of the same frequency. Because of interference, unintended side-effects can be observed otherwise (spaghettification, additional ears, etc.).

For instance, the following figure illustrates 5 teleports A, B, C D and E along with their operating frequencies. The links show all possible pairs between which mice can travel directly.



As one can observe, to travel between a given pair of teleports, one sometimes has to travel through some amount of intermediate teleports. For instance, when travelling from B to C, one needs to change at A (or D). Formally, we define $d(x, y)$ as the distance between two teleports x and y to be the minimum number of teleportations needed to get from one to the other. In the aforementioned case, $d(B, C) = 2$.

The alien mice are very proud of their network and claim that it is quite efficient: one does not need to use too many intermediate teleports to travel between any two locations.

An objective measure for this is the maximal shortest distance between any two teleporters. In the example above $d(A, E) = 3$, the maximum shortest distance between two teleporters.

Subtask 1: Solve some Network (10 Points)

We have 7 teleports with the following operating frequencies: $A = (6, 8, 3)$, $B = (4, 7, 0)$, $C = (5, 6, 2)$, $D = (1, 7, 3)$, $E = (2, 4, 5)$, $F = (6, 8, 1)$, $G = (4, 8, 0)$. List all pairs of teleports between which Stofl can *directly* travel.

Additionally, find a pair of teleporters x and y in this teleportation network such that $d(x, y)$ is the maximum shortest distance.

Subtask 2: Network with Maximum Distance (45 Points)

Stofl wonders whether there may be a network where the maximum distance between two teleports is large.

Task Give a list of operating frequences of at least 2 teleports and indicate the two end points x and y, such that the distance between x and y is as large as possible.

Scoring For a network with maximum distance ℓ, you will get $\min(5\ell - 15, 45)$ points. That is, if the maximum distance is 12 or more, you will be awarded full points.

Example The teleports in the above example are not that far away. The maximum distance of 3 teleportations is between teleports A and E. This network would be worth 0 points. :)

Subtask 3: Bound for Maximum Distance (45 Points)

Stofl struggles to imagine a network, where the maximum distance would be 2017 and starts to think that such distance is impossible to achieve.

Task Give a proof that the maximum distance in a network of teleports is less than 2017. Try to improve the bound as much as possible.

Scoring Any completely correct proof with arbitrary constant less than 2017 will be awarded 20 points. If the constant is at most two times the optimum, it will get 25 points. Other bounds will be scaled accordingly.

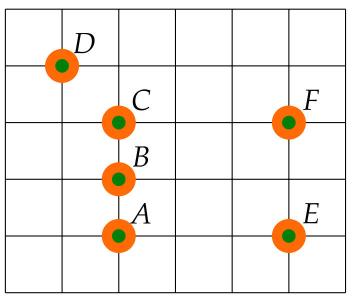
Carrot Garden

Mouse Stofl has started to grow carrots in his garden. But mouse Stofl is a scientist and growing stuff gets boring quickly. He shifted his attention towards the parasites in his garden. Mouse Stofl believes to have found a new kind of parasite and has named this new species after himself: "Stoffulus". What's interesting about Stoffulus is his spreading pattern. An infection always starts with two infected carrots. A healthy carrot gets infected exactly then when her distance to an already infected carrot is smaller or equal than the distance between any two already infected carrots.

Mouse Stofl now wants to know the following: If he infects two carrots with Stoffulus, the parasite will spread until he can't reach any new carrots. We will call this state the "final stage". The carrots infected in the final stage obviously depend on which two carrots were chosen at the beginning. Mouse Stofl now wants to know how many different final stages there exist for his garden. But there are many possibilities to choose two carrots to infect at the beginning ($\binom{N}{2} = \frac{N(N-1)}{2}$ to be exact). Mouse Stofl doesn't want to conduct this many experiments and asks you for help.

Formal Description Given n different points $p_1, \ldots, p_n$ in the plane (Stofl's garden). Each experiment starts with two infected points: p_a, p_b. A point p_x gets infected if there exists three (not necessarily different) infected points p_v, p_w, p_y such that $\text{dist}(p_x, p_y) \leq \text{dist}(p_v, p_w)$[1]. As soon as there is no point left, that could be infected, we reach the final stage. For each pair p_a and p_b there is a final stage. How many *different* final stages are there? Two final stages differ if there is at least one carrot in one final stage but not the other one.

Example Look at the example below where all carrots are placed on a grid (the circles stand for the carrots). There are two different final stages: (a) everything and (b) only E and F.

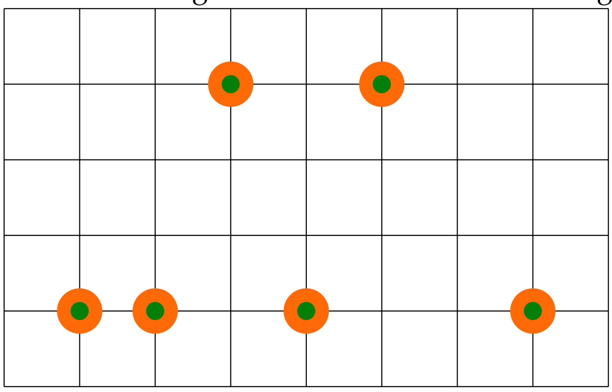


Let's look what happens if A and B are infected. Because of $d(A, B) \geq d(B, C)$, C gets infected as well. Then, because of $d(A, C) \geq d(C, D)$, we get D. Lastly, the infection jumps over to the right: $d(A, D) \geq d(A, E)$ and $d(A, D) \geq d(C, F)$, and all carrots are infected.

[1]dist is the Euclidean distance $\text{dist}((x_1, y_1), (x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Subtask 1: Solve the Example (10 points)

For this example, we assume as well, that all carrots are placed on a grid. How many different final stages are there? Write down a single integer number.



Subtask 2: Develop an Algorithm (90 points)

Develop an algorithm for Stofl that takes a list $L = [(x_1, y_1), \ldots, (x_n, y_n)]$ of points as input and outputs one single integer, the number of different final stages.

As usual, write down some *pseudocode*, the *runtime complexity*, *memory usage* of your algorithm and explain its *correctness*.

Cake Protection Agency

Mouseland has multiple elite police forces with different missions, one of the (arguably) most important ones is the CPA, the cake protection agency. They are tasked with protecting all the important cakes in the land from vicious attacks.

The CPA just got the word of a very dangerous cake terrorist, mouse Gehr, plotting an attack. Apparently he plans to leave his home, walk to the cake and throw it to the ground. The only way for the CPA agents to stop him is to either arrive at the cake earlier than terrorist Gehr (and move the cake to a more secure location), or to intercept and detain him at an intersection before reaching the cake.

Your task will be to help the noble agents of the CPA trying to stop cake terrorist Gehr.

Formal Description Assume mouseland is given as a *weighted*, directed graph $G = (V, E)$ with $n + 3$ vertices $V = \{v_{\text{cpa}}, v_{\text{gehr}}, v_{\text{cake}}, v_1, \dots v_n\}$, the starting base of the CPA agents, the initial location of mouse Gehr, the location of the cake and the remaining intersections in mouseland. From each vertex you can reach the cake (it's not a lie). An edge $e = (u, v, t) \in E$ connects $u \in V$ to $v \in V$ with integer travel time $t \geq 0$. Note that all roads in mouseland are one way, so the edge (u, v, t) can only be travelled from u to v and not v to u (there might be an additional edge (v, u, t') in the graph to travel in the opposite direction). The CPA agents can stop mouse Gehr if one arrives at the cake strictly earlier than mouse Gehr or if they occupy the same node $v \neq v_{\text{cake}}$ at the same time.

Subtask 1: Latest Starting Time (15 points)

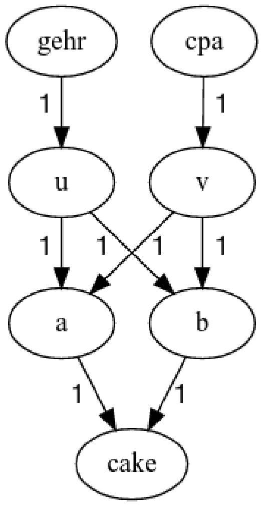
Mouse Gehr was not careful enough to protect his secret plan and his exact route $p = (v_{\text{gehr}}, \dots, v_{\text{cake}})$ to the cake as well as his (integer) start time T got leaked. Design an algorithm which finds the latest integer time T' at which a CPA agent has to leave the CPA base v_{cpa} to stop mouse Gehr by either arriving earlier at the cake or intercepting him along his route. Argue why your algorithm is correct, write down pseudocode, analyze its asymptotic runtime and its memory usage.

Subtask 2: Plenty of Agents (35 points)

Mouse Gehr was more careful and his route didn't leak. Assume he leaves his home at time T and the CPA can send out arbitrarily many agents. Find the latest integer time T', when those agents have to be dispatched in order to stop mouse Gehr. Argue why your algorithm is correct and analyse its runtime complexity.

Also note that mouse Gehr knows where all deployed CPA agents are at any time and vice versa. Moreever, whenever a CPA agent and Mouse Gehr have to make a decision (which outgoing edge to follow) at the exact same time, then mouse Gehr decides first and the CPA agent *will know* which one he chose.

This is illustrated in the following graph. Assume all edge weights are 1, in that case *one* CPA agent leaving at the same time as mouse Gehr will always catch him, because the optimal strategy for both is going to be u (for mouse Gehr) and v (for the CPA agent) in the first step. After that, mouse Gehr has to decide first between a and b, and the CPA agent can just mirror his choice to meet and detain him at the respective node.



Subtask 3: One Agent (50 points)

Assume the same setup as in subtask B, however, the CPA only has one agent available. Again determine the latest integer time T' when he needs to leave the CPA base v_{cpa} in order to stop mouse Gehr. Argue why your algorithm is correct, write down pseudocode and analyse its runtime complexity and its memory usage.