



## Algorithms & Data Structures

## Exercise sheet 11

## HS 24

The solutions for this sheet are submitted on Moodle until 8 December 2024, 23:59.

Exercises that are marked by \* are challenge exercises. They do not count towards bonus points.

You can use results from previous parts without solving those parts.

### Exercise 11.1 *Multiplicatively shortest path (1 point).*

Let  $G = (V, E)$  be a directed graph, with weights  $w_e \geq 1$  for  $e \in E$ . The *multiplicative weight* of a path  $P \subseteq E$  is

$$\text{mw}(P) := \prod_{e \in P} w_e \quad (\text{i.e., the product of the weights of the edges in } P).$$

Given  $s, t \in V$ , give an algorithm that computes the multiplicative weight of a shortest path from  $s$  to  $t$  (i.e., a path  $P$  from  $s$  to  $t$  for which  $\text{mw}(P)$  is minimized). You may assume that there is at least one path from  $s$  to  $t$ .

The runtime of your algorithm should be at most  $O((n + m) \cdot \log n)$ . You should argue briefly why your algorithm is correct and why it achieves the desired runtime.

**Hint:** You may use any algorithm you have seen in the lecture as a subroutine and use any runtime bounds you have seen for them without proof.

**Hint:** For all  $a, b > 0$ , we have  $\log(a \cdot b) = \log(a) + \log(b)$ .

### Exercise 11.2 *Rotating weights (1 point).*

Let  $G = (V, E)$  be a complete graph with  $n$  vertices. Enumerate the vertices as  $v_0, \dots, v_{n-1}$ . Let  $c_0 : E \rightarrow \mathbb{R}$  be an edge weight function. When we are walking on this graph, the edge weights 'rotate', meaning at step  $t$  along a walk, we will have the edge weight function  $c_t : E \rightarrow \mathbb{R}$  where  $c_t(\{v_i, v_j\}) = c_0(\{v_{i+t}, v_{j+t}\})$  and  $i + t, j + t$  are taken modulo  $n$ . Formally, given a walk  $(v_{i_0}, v_{i_1}, \dots, v_{i_k}, v_{i_{k+1}})$ , its weight is  $c_0(\{v_{i_0}, v_{i_1}\}) + c_1(\{v_{i_1}, v_{i_2}\}) + \dots + c_k(\{v_{i_k}, v_{i_{k+1}}\})$

Suppose that there is no way to construct a cycle which has negative weight. Describe an algorithm that finds a shortest walk from  $v_0$  to  $v_1$  in time  $O(n^5)$ . Prove that your algorithm is correct, and achieves the desired runtime.

**Remark.** Recall that given two positive integers  $a$  and  $b$ ,  $a$  modulo  $b$  is the remainder of the division of  $a$  by  $b$ . For example, assuming  $n > 1$ , we have  $(n + 1)$  modulo  $n$  equal to 1.

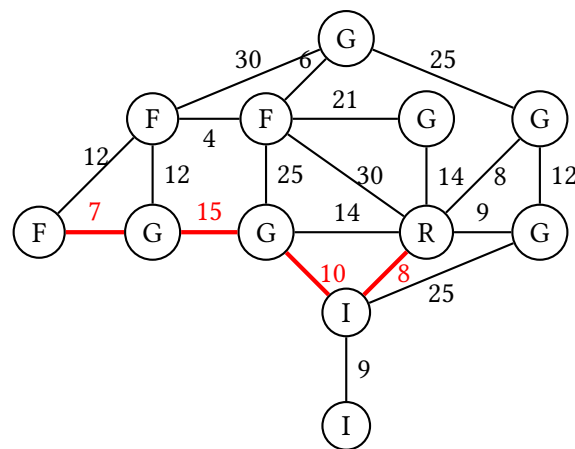
**Exercise 11.3** *Language Hiking.*

Alice loves both hiking and learning new languages. Since she moved to Switzerland, she has always wanted to discover all four language regions of the country in a single hike – but she is not sure whether her week of vacation will be sufficient.

You are given a graph  $G = (V, E)$  representing the towns of Switzerland. Each vertex  $V$  corresponds to a town, and there is an (undirected) edge  $\{v_1, v_2\} \in E$  if and only if there exists a direct road going from town  $v_1$  to town  $v_2$ . Additionally, there is a function  $w : E \rightarrow \mathbb{N}$  such that  $w(e)$  corresponds to the number of hours needed to hike over road  $e$ , and a function  $\ell : V \rightarrow \{G, F, I, R\}$  that maps each town to the language that is spoken there<sup>1</sup>. For simplicity, we assume that only one language is spoken in each town.

Alice asks you to find an algorithm that returns the walking duration (in hours) of the shortest hike that goes through at least one town speaking each of the four languages.

For example, consider the following graph, where languages appear on vertices:



The shortest path satisfying the condition is marked in red. It goes through one R vertex, one I vertex, two G vertices and one F vertex. Your algorithm should return the cost of this path, i.e., 40.

- (a) Suppose we know the order of languages encountered in the shortest hike. It first goes from an R vertex to an I vertex, then immediately to a G vertex, and reaches an F vertex in the end, after going through zero, one or more additional G vertices. In other terms, the form of the path is RIGF or RIG...GF. In this case, describe an algorithm which finds the shortest path satisfying the condition, and explain its runtime complexity. Your algorithm must have complexity at most  $O((|V| + |E|) \log |V|)$ .

**Hint:** Consider the new vertex set  $V' = V \times \{1, 2, 3, 4\} \cup \{v_s, v_d\}$ , where  $v_s$  is a ‘super source’ and  $v_d$  a ‘super destination’ vertex.

- (b) Now we don’t make the assumption in (a). Describe an algorithm which finds the shortest path satisfying the condition. Briefly explain your approach and the resulting runtime complexity. Your algorithm must have complexity at most  $O((|V| + |E|) \log |V|)$ .

**Hint:** Consider the new vertex set  $V' = V \times \{0, 1\}^4 \cup \{v_s, v_d\}$ , where  $v_s$  is a ‘super source’ and  $v_d$  a ‘super destination’ vertex.

<sup>1</sup>G, F, I and R stand for German, French, Italian, and Romansh respectively.

**Exercise 11.4** *Rail racer (1 point).*

You are designing new routes for freight trains to ship cargo from Zurich to Zermatt. The railway network of Switzerland has many train stations (including Zurich and Zermatt). For simplicity, we assume there is at most one direct railway (i.e. not passing a third station) between any two stations and its length is a multiple of 10 km; also, the trains can travel on both directions of all railways. To guarantee safety, checkpoints are installed at every train station as well as at every 10 km on the railways. (For example, if there is railway of 40 km between station A and station B, then there will be 3 checkpoints on this railway in addition to the 2 checkpoints at station A and station B.) At each checkpoint, there is a speed limit: 50, 100, or 150 km/h. The trains should not exceed the speed limits at any checkpoint. The trains have three speed settings of 50, 100 and 150 km/h. The trains can increase or decrease its speed by 50 km/h linearly over a 10 km stretch of rail. The trains begin the journey from Zurich at 50 km/h and must end in Zermatt at 50 km/h. (We ignore the acceleration phase from 0 to 50 km/h and the deceleration phase from 50 to 0 km/h.)

Calculate the shortest time the trains need to get from Zurich to Zermatt.

- (a)\* Calculating the acceleration can be done via the formula  $a = (v_1^2 - v_0^2)/(2d)$  where  $v_0, v_1$  are the initial and final speeds and  $d$  is the distance traveled. Using this formula, find the time it takes to traverse a 10km stretch of rails depending on your starting speed (50, 100 or 150 km/h) and also depending on whether you stay at a constant speed, or you accelerate/decelerate to the higher/lower speed.

**Hint:** The relationship between the time taken, acceleration and initial and final speeds is given by  $v_1 = v_0 + a \cdot t$ .

- (b) Model the problem as a graph problem such that you can directly apply one of the algorithms in the lecture, without modifications to the algorithm:
- (1) Describe your graph. What are the vertices, what are the edges and the weights of the edges?
  - (2) What is the graph problem that we are trying to solve?
  - (3) Solve the problem using an algorithm discussed in the lecture (without modification). a

**Note:** If you didn't solve part (a)\*, you can assume the time it takes to travel 10 km with starting speed  $v_0$  and ending speed  $v_1$  is  $c/(v_0 + v_1)$  for some positive integer  $c$ .

**Exercise 11.5** *Modernization of the Deutsche Bahn.*

The Deutsche Bahn (DB) has  $n$  train stations and  $m$  train tracks connecting them, so that from one station you can reach all other  $n - 1$  stations. To save money, not all train tracks are actively maintained. However, the current state leads to a great number of problems, such as high delays and general unreliability, and this is very annoying for some ETH students traveling home for Christmas.

The DB wants to modernize some of the train tracks so that every two stations are connected by modernized tracks. Unfortunately, modernized tracks require frequent investments to maintain. For this exercise, we assume that the price for modernizing a track is negligible and we only consider the yearly maintenance cost. This cost is proportional to the length of the track. The DB modernizes a subset of the tracks such that the entire yearly maintenance cost is as small as possible.

After some time of operation, the DB notices that they also need a new signal at every modernized track to fully operate it. Again, we assume that the building cost is negligible and we don't consider it in this exercise. These signals also have a yearly maintenance cost  $k$ , which is the same for each signal.

Now, the DB asks you: in order to minimize the overall yearly maintenance cost (for both tracks and signals combined), whether they can just maintain the set of modernized tracks they already have and build a signal on each modernized track? Or do they have to modernize some new tracks (and potentially shut down some already modernized tracks) in order to be able to achieve the minimum yearly maintenance cost for both signals and tracks?

Decide whether they can keep the already maintained tracks or not. If so, explain your reasoning, otherwise provide a counterexample that shows that it might be necessary to modernize new tracks. Recall that we assume that the cost of modernizing a track is negligible and that we are only interested in minimizing the yearly maintenance cost.