Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

Departement of Computer Science
Johannes Lengler, Markus Püschel, David Steurer
Kasper Lindberg, Kostas Lakis, Lucas Pesenti, Manuel Wiedmer

1 December 2025

Algorithms & Data Structures — Exercise sheet 11 — HS 25

The solutions for this sheet are submitted on Moodle until 7 December 2025, 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 *Shortest paths with cheating* **(1 point)**.

Let $G = (V, E)$ be a weighted, directed graph with weights $c : E \to \mathbb{R}_{\geq 0}$. We consider a variation of the shortest path problem in G, where we are allowed to 'cheat' by setting a certain number of weights to 0. Formally, for $k \in \mathbb{N}$, we write C_k for the set of all weight functions $\gamma : E \to \mathbb{R}_{\geq 0}$ on G with $\gamma(e) \neq c(e)$ for at most k edges $e \in E$.[1]

Given $s, t \in V$, we wish to find a path $P = (v_1 = s, v_2, \ldots, v_\ell = t)$ in G which minimizes:

$$c_k(P) := \min_{\gamma \in C_k} \gamma(P), \text{ where } \gamma(P) := \sum_{i=1}^{\ell-1} \gamma\big((v_i, v_{i+1})\big).$$

We call such a path a 'shortest path from s to t with k cheats.'

Recall that a naive implementation of Dijkstra's algorithm finds the length of a shortest path in a weighted graph (without cheating) in time $O(|V|^2)$.

(a) Describe an algorithm which finds the length of a shortest path from s to t with k cheats in time $O(|E|^k \cdot |V|^2)$.

Address the following elements:

1) The graph that you use. This includes: its vertex set and edge set; whether edges are directed or not; whether vertices or edges are weighted and if so, their weight.
2) The algorithm that you apply to this graph. You can use the algorithms covered in the lecture material as subroutines, and you can use their running time bounds without proof.
3) How you extract the solution from (the output of) your algorithm.
4) The total running time of your proposed algorithm (including construction the graph and extracting the solution), with a short justification.

Hint: *Apply Dijkstra's algorithm to G several times, for different weight functions.*

(b) Describe an algorithm which finds the length of a shortest path from s to t with k cheats in time $O\big((k|V|)^2\big)$.

Address the following elements:

[1] We assume that $|E| \geq k$.

1) The graph that you use. This includes: its vertex set and edge set; whether edges are directed or not; whether vertices or edges are weighted and if so, their weight.
2) The algorithm that you apply to this graph. You can use the algorithms covered in the lecture material as subroutines, and you can use their running time bounds without proof.
3) How you extract the solution from (the output of) your algorithm.
4) The total running time of your proposed algorithm (including construction the graph and extracting the solution), with a short justification.

Hint: *Construct a new graph $G' = (V', E')$ whose vertex set V' consists of $k+1$ copies of V. Choose the edges E' and weights c' in a clever way, and apply Dijkstra's algorithm to G'.*

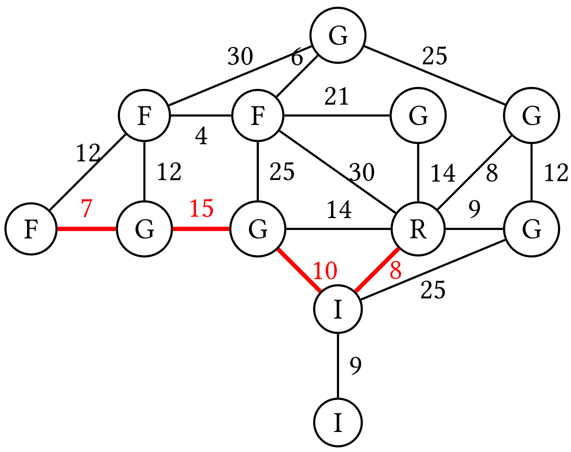
Exercise 11.2 *Language Hiking.*

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 $w : E \to \mathbb{N}$ such that $w(e)$ corresponds to the number of hours needed to hike over road e, and a function $\ell : V \to \{\mathrm{G}, \mathrm{F}, \mathrm{I}, \mathrm{R}\}$ that maps each town to the language that is spoken there[2]. 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, af-

[2] G, F, I and R stand for German, French, Italian, and Romansh respectively.

ter 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 $O((|V| + |E|) \log |V|)$.

Address the following elements:

1) The graph that you use. This includes: its vertex set and edge set; whether edges are directed or not; whether vertices or edges are weighted and if so, their weight.
2) The algorithm that you apply to this graph. You can use the algorithms covered in the lecture material as subroutines, and you can use their running time bounds without proof.
3) How you extract the solution from (the output of) your algorithm.
4) The total running time of your proposed algorithm (including construction the graph and extracting the solution), with a short justification.

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 $O((|V| + |E|) \log |V|)$. Address the following elements:

1) The graph that you use. This includes: its vertex set and edge set; whether edges are directed or not; whether vertices or edges are weighted and if so, their weight.
2) The algorithm that you apply to this graph. You can use the algorithms covered in the lecture material as subroutines, and you can use their running time bounds without proof.
3) How you extract the solution from (the output of) your algorithm.
4) The total running time of your proposed algorithm (including construction the graph and extracting the solution), with a short justification.

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.*

Exercise 11.3 *Rail racer.*

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: *T*he 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.

Address the following elements:

1) The graph that you use. This includes: its vertex set and edge set; whether edges are directed or not; whether vertices or edges are weighted and if so, their weight.
2) The algorithm that you apply to this graph. You can use the algorithms covered in the lecture material as subroutines, and you can use their running time bounds without proof.
3) How you extract the solution from (the output of) your algorithm.
4) The total running time of your proposed algorithm (including construction the graph and extracting the solution), with a short justification.

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 11.4 *Driving from Zurich to Geneva* **(1 point)**.

Bob is currently in Zurich and wants to visit his friend that lives in Geneva. He wants to travel there by car and wants to use only highways. His goal is to get to Geneva as cheap as possible. He has a map of the cities in Europe and which ones are connected by highways (in both directions). For each highway connecting two cities he knows how much fuel he will need for this part (depending on the length, condition of the road, speed limit, etc.) and how much this will cost him. This cost might be different depending on the direction in which he travels. Furthermore, for some connections between two cities, he has the option to take a passenger with him that will pay him a certain amount of money. Again this might be different depending on the direction he travels. We assume that this option is only available to him between cities directly connected by a highway and that the passengers want to travel the direct road and would not agree to making a detour. Also Bob has a small car, so he can only take at most one passenger with him. It is possible that he gains more money from this than he has to pay for the fuel between two given cities but we assume that he has no way to gain an infinite amount of money, i.e. there is no round-trip from any city that earns him money.

(a) Model the problem as a graph problem.

Address the following elements:

1) The graph that you use. This includes: its vertex set and edge set; whether edges are directed or not; whether vertices or edges are weighted and if so, their weight.
2) The algorithm that you apply to this graph. You can use the algorithms covered in the lecture material as subroutines, and you can use their running time bounds without proof.
3) How you extract the solution from (the output of) your algorithm.

4) The total running time of your proposed algorithm (including construction the graph and extracting the solution), with a short justification.

Exercise 11.5 *Ancient Kingdom of Macedonia* **(1 point)**.

The ancient Kingdom of Macedonia has n cities and m roads connecting them, such that from any city you can reach all other $n-1$ cities. Each road has a given length. The existing roads are Roman roads: they are stone-paved and require no maintenance. A new, much faster vehicle, the Tesla Carriage, has been invented. It can only travel on asphalt roads, and each asphalt road of length ℓ has a yearly maintenance cost proportional to ℓ.

In the first year of his reign, Philip II decides to modernize the kingdom by turning some of the existing Roman roads into asphalt roads, so that every two cities can be reached with a Tesla Carriage (i.e., with asphalt roads). He chooses a set of roads that satisfies this requirement while minimizing the total yearly maintenance cost of the asphalt roads.

In the second year, to fight robberies, Philip II decides to place a checkpoint on every asphalt road. Each checkpoint has the same fixed yearly cost k (the same for every asphalt road), in addition to the existing maintenance cost of the asphalt road itself. Before building the checkpoints, Philip II is allowed, if he wishes, to change which roads are asphalt: he may remove asphalt from some roads and add asphalt to others. However, he must still ensure that every two cities can be reached using only asphalt roads. His goal is to minimize the total yearly maintenance cost. Does he need to reconsider which roads are asphalt, or can he keep the same set of asphalt roads that he built in the first year? Prove your answer or provide a counterexample.