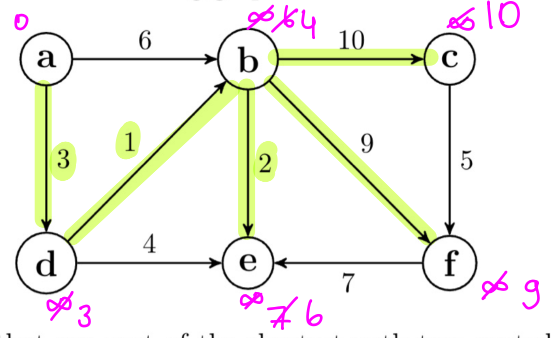
Dijkstra's Algo - Exam Questions

/ 2 P

f) *Shortest Path Tree:* Consider the following graph:



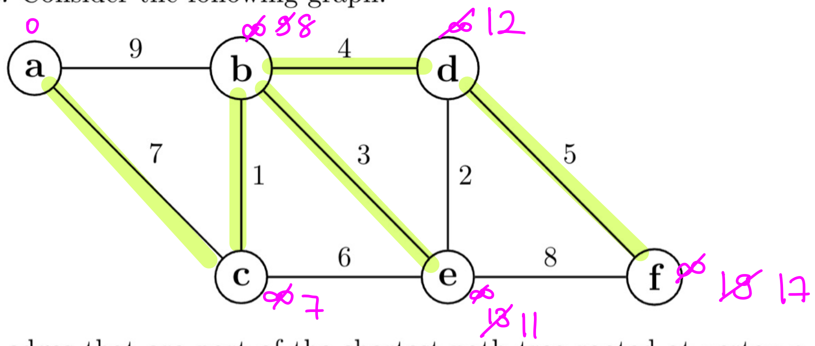
i) Highlight the edges that are part of the shortest-path tree rooted at vertex a (i.e., the output of Dijkstra's algorithm if we were to start from vertex a).

ii) Does the above graph have a topological ordering? If yes, write down one topological ordering. If no, give an argument.

Yes , a, d, b, c, f, e

/ 2 P

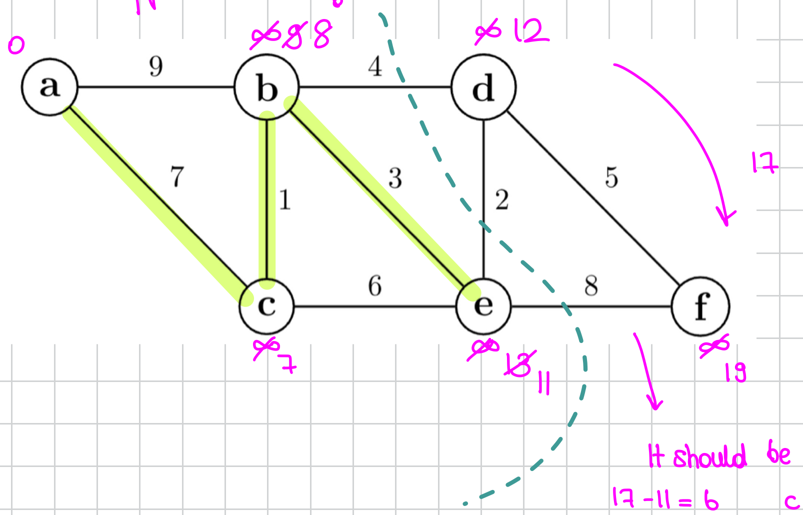
f) *Shortest Path Tree:* Consider the following graph:



i) Highlight the edges that are part of the shortest-path tree rooted at vertex a (i.e., the output of Dijkstra's algorithm if we were to start from vertex a).

ii) Write out all positive integers x such that we could replace the weight 8 of the edge $\{e, f\}$ in the above graph by x, such that the edge would be in at least one shortest-path tree rooted at a of the resulting graph.

what happened there?



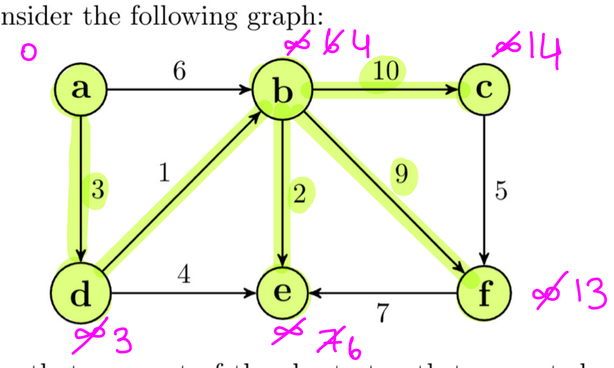
It should be ≤ 17 !

17 - 11 = 6 $c(e,f) \leq 6$!

1, 2, 3, 4, 5, 6

Dijkstra's Algo - Exam Questions

/ **2 P** f) *Shortest Path Tree:* Consider the following graph:

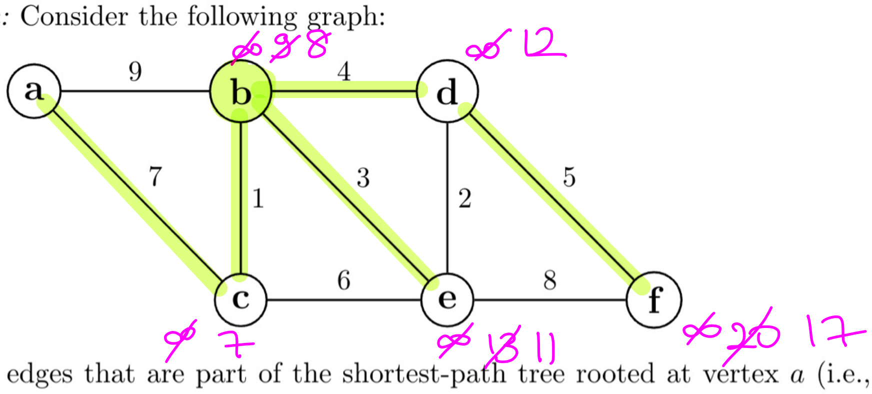


i) Highlight the edges that are part of the shortest-path tree rooted at vertex a (i.e., the output of Dijkstra's algorithm if we were to start from vertex a).

ii) Does the above graph have a topological ordering? If yes, write down one topological ordering. If no, give an argument.

Yes ,

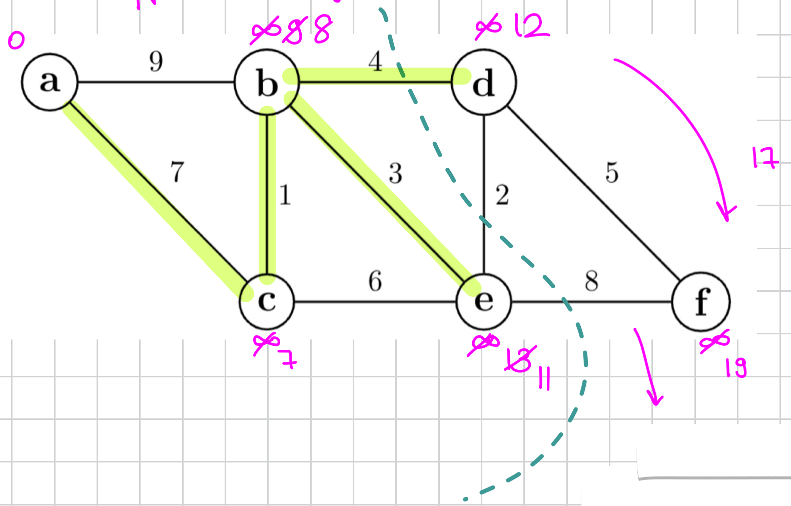
/ **2 P** f) *Shortest Path Tree:* Consider the following graph:



i) Highlight the edges that are part of the shortest-path tree rooted at vertex a (i.e., the output of Dijkstra's algorithm if we were to start from vertex a).

ii) Write out all positive integers x such that we could replace the weight 8 of the edge $\{e, f\}$ in the above graph by x, such that the edge would be in at least one shortest-path tree rooted at a of the resulting graph.

what happened there?



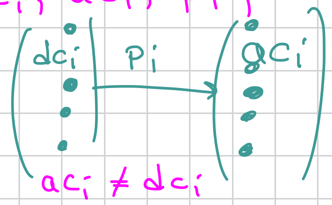
x →

Graph Modelling

/ 7 P a) Alice lives in Zurich and wants to visit her grandma in Stockholm. Since she has $n \in \mathbb{N}$ full weeks of vacation time from September 1st, she would like to use some of this time travel by train from Zurich to Stockholm and discover nice cities along the way. However, as a student, Alice may have a lot of time, but not a lot of money. Hence, while she does not mind spending long hours in the train, she would like to minimize her costs. You are tasked with helping her find the cost of the cheapest route from Zurich to Stockholm.

You are given a timetable composed of N rows $(r_1, \ldots, r_N)$, each representing one possible (direct) train journey. Let $\mathcal{C}$ be the set of cities. Each row r_i is of the form $r_i = (dt_i, at_i, dc_i, ac_i, p_i)$ where $dt_i \in \mathbb{N}$ is the departure time of the train, $at_i \in \mathbb{N}$ its arrival time, $cd_i \in \mathcal{C}$ its departure city, $ac_i \neq dc_i \in \mathcal{C}$ its arrival city, and $p_i \in \mathbb{N}$ the price of the journey in CHF. All departure and arrival times are expressed in minutes from September 1st, midnight. There are $D = 1440$ minutes in a day. You are guaranteed that the timetable allows Alice to travel from Zurich to Stockholm in less than n weeks, or $7 \cdot D \cdot n$ minutes. Note that as a Youth Hostel Premium Member, Alice does not need to pay anything to stay in the various cities she travels through.

$$N \text{ rows } \left\{ \begin{matrix} r_1 \\ \vdots \\ r_N \end{matrix} \right. \longrightarrow r_i = (\underset{\text{departure time}}{dt_i}, \underset{\text{arrival time}}{at_i}, \overset{\text{departure city}}{dc_i}, \overset{\text{arrival city}}{ac_i}, \overset{\text{price of journey}}{p_i})$$



$ac_i \neq dc_i$

C : set of cities

minute 0 - - - minute $7 \cdot D \cdot n$

September 1st midnight

1440 mins in a day

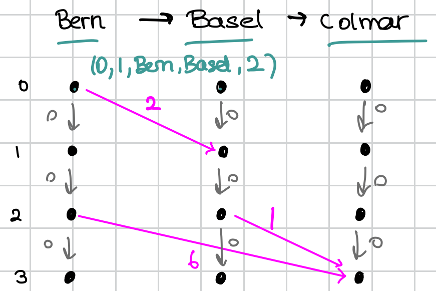
n weeks of vacation

/ 7 P b) Alice will likely use trains of a famous railway company on her way, and these trains are notoriously unreliable. In fact, each train can be cancelled arbitrarily. Being aware of this situation, Alice wants to plan for the worst and know how much more expensive the journey can become if trains are cancelled.

Alice has now selected her preferred journey. She will visit cities $c_1 = $ Zurich, $c_2, \ldots, c_{T+1} = $ Stockholm in this order, with an expected total cost t. The timetable is the same as before, but now, even if trains are cancelled, Alice will always follow the same routes $c_1 \to c_2$, ..., $c_T \to c_{T+1}$.

For each of the T journeys, Alice buys the ticket at the departure station shortly before the train starts. When buying a ticket, Alice can be sure that her next train will work, but she never knows whether further connections will behave as expected. The only thing that the railway company **does** guarantee to Alice is that she will be able to reach Stockholm from Zurich within n weeks eventually.

$\overset{\text{Zurich}}{c_1}, \overset{\text{Basel}}{c_2}, c_3 \ldots, c_T, \overset{\text{Stockholm}}{c_{T+1}}$ expected total cost t



i) Model the problem as a graph problem, so that you can solve it using an algorithm from the lecture. Describe the set of vertices, the set of edges and the weights. How many vertices and edges does your graph have? What is the corresponding graph problem?

$G = (V, E, w)$ is a weighted, directed Graph.

$V = \{(c, t) \mid c \in C, \; 0 \leq t \leq 7 \cdot D \cdot n\}$ $\qquad |V| = |C| \cdot (7Dn+1)$

$E = \{((dc_i, dt_i), (ac_i, at_i)) \mid 1 \leq i \leq N\}$
$\cup$
$\{((c, t), (c, t')) \mid c \in C, \; 0 \leq t < t' \leq 7 \cdot D \cdot n\}$ $\qquad |E| = N + (7 \cdot D \cdot n)+1) \cdot |C|$ edges

$w(((dc_i, dt_i), (ac_i, at_i))) = p_i \quad \forall 1 \leq i \leq N$ $\qquad w(((c,t), (c, t+1))) = 0 \quad \forall c \in C, \; 0 \leq t < 7 \cdot D \cdot n$

Graph Problem : Cost of the shortest path from (Zürich, 0) to (Stockholm, $7 \cdot D \cdot n$)

ii) Which algorithm from the lecture can you use to solve this graph problem? Justify why you can use this algorithm, and state its asymptotic running time (with Θ) in terms of the number of vertices $|V|$ and edges $|E|$ in the graph.

Dijkstra (weighted, nonnegative edge costs)

in $\Theta((|V| + |E|) \cdot \log |V|)$

i) Model the problem as a longest path problem in a directed acyclic graph. Describe the set of vertices, the set of edges and the weights. Prove that your directed graph is acyclic and explain why the longest path problem you propose provides the right solution to the problem.

$C' = (c_1, c_2, \ldots, c_{T+1})$ $\qquad G' = (V', E', w)$ is a weighted, directed graph.

$V' = \{(c, t) \mid c \in C', \; 0 \leq t \leq 7 \cdot D \cdot n\}$ $\qquad c_j \to c_{j+1}$

$E' = \{((dc_i, dt_i), (ac_i, at_i)) \mid 1 \leq i \leq N, \; \exists\, 1 \leq j < T. \; dc_i = c_j \wedge ac_i = c_{j+1}\}$
$\cup$
$\{((c, t), (c, t')) \mid c \in C, \; 0 \leq t < t' \leq 7 \cdot D \cdot n\}$

w just like in a

Graph Problem : cost of the longest path from (Zurich, 0) to (Stockholm, $7 \cdot D \cdot n$) in G'

Worst scenario would be that only the most expensive journey remaining uncancelled!

ii) Design an efficient algorithm that determines how much more expensive Alice's journey can become. In order to get full points, your algorithm should have $O(|V||E|)$ runtime.

Hint: *What happens if you replace every edge with cost w by an edge with cost $-w$? Recall that, as proven in Task 2, a longest path always exists.*

- Compute $G'' = (V', E', w')$ where $w'(e) = -w(e)$ for all $e \in E$
- Bellman-Ford starting from (Zurich, 0) (to obtain $d_{G''}((\text{Zurich}, 0), (\text{Stockholm}, 7 \cdot D \cdot n))$)
- Return $-d_{G''}((\text{Zurich}, 0), (\text{Stockholm}, 7 \cdot D \cdot n))$ $\quad -d_{G''} - t$

Bellman-Ford runtime: $O(|V| \cdot |E|)$

(No Dijkstra : G'' has negative weights, but
contains no cycles
Bellman Ford without cycle detection is enough)