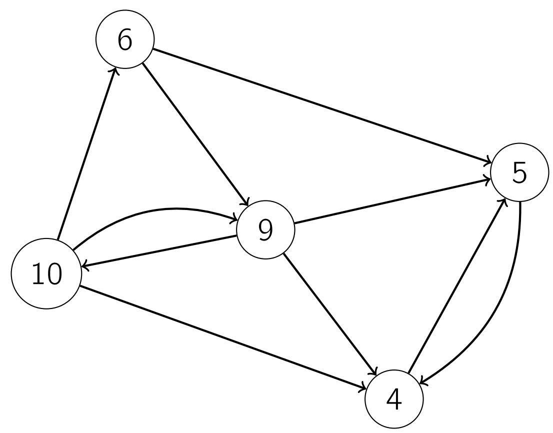
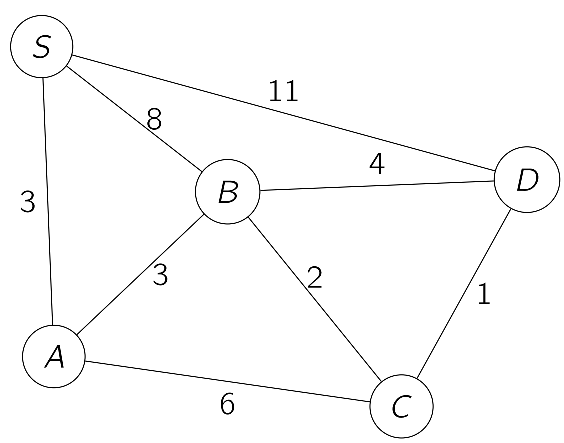
CS 161 (Stanford, Fall 2025) Section 5

1 Algorithm Practice

1. In the given graph, the node labels represent the finish times from running depth-first search. Which node would the next DFS call begin from when running Kosaraju's algorithm? Perform this DFS (with edges reversed) to find the find the strongly connected components of the graph.

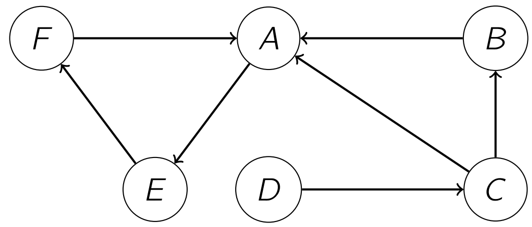


2. Perform Dijkstra's shortest path algorithm from source S on the graph below, and update the $d[v]$ values for each iteration in the table.



2 Strongly Connected Components

Consider the directed graph below for parts 1 and 2:



(a) How many strongly connected components does this graph have?

(b) What is the minimum number of directed edges to add to this graph to make all the vertices strongly connected?

(c) Assume you have two vertices u and v in a directed graph where there exists an edge from u to v. Which one of the following is incorrect about u and v?

(A) u and v can be in the same SCC.

(B) u and v can be in different SCCs.

(C) If u's DFS finish time is less than v's DFS finish time then u and v are in the same SCC.

(D) u's DFS finish time is always greater than v's DFS finish time.

3 Edsger's Apfelstrudel

You are eating at a cozy little restaurant which serves a *prix fixe* menu of $k+1$ courses, with several available choices for each course. Each dish belongs to exactly one course (e.g., risotto can only be ordered as an appetizer, not a main), and you are effectively indifferent between most of the items on the menu (because they are all so tasty), but the main draw of this particular restaurant is that they serve a delicious 'bottomless' dessert: their world-famous Viennese-style apple strudel. They have an unlimited supply of this apple strudel, but each serving will still cost you \$1. The restaurant also has a few interesting rules:

(a) You must finish your current dish before ordering another.

(b) Each dish after the first course depends on what you ordered in the previous course, e.g., you can only order salmon for your main if you ordered a Caesar salad or chicken noodle soup for the previous course. You are told on the menu exactly what these restrictions are before you order anything.

(c) Most importantly, you are not allowed to have their unlimited dessert unless you finish one dish from each of the first k courses!

You are told the cost of each item in each course on the menu, and you plan your meal with a twofold goal: to be able to order the strudel, but also to save as much money as possible throughout the first k courses so that you have more money to spend on the unlimited dessert. Design an algorithm to find the smallest amount of money you can spend on the first k courses and still order the 'bottomless' strudel. If you would like, you may assume the very first course has exactly one choice (e.g., a single complimentary leaf of spinach that costs 0 dollars). Give an $O(k\log(k) + m)$ time algorithm, where m is the number of pairs of dishes for which one dish depends on the other.

4 Finding a target vertex

Let $G = (V, E)$ be a directed acyclic graph. Say that a vertex v is a *target* if, for all $u \in V$, there is a path from u to v. Given an algorithm that runs in time $O(n + m)$ that either finds a target vertex, or else returns `None` if no target vertex exists.

5 High Speed Cable Internet

Algorithmia, an internet service provider, has a new high speed cable internet technology that will require new cable installation. They will install these new cables on the currently existing network of cables but it will be costly.

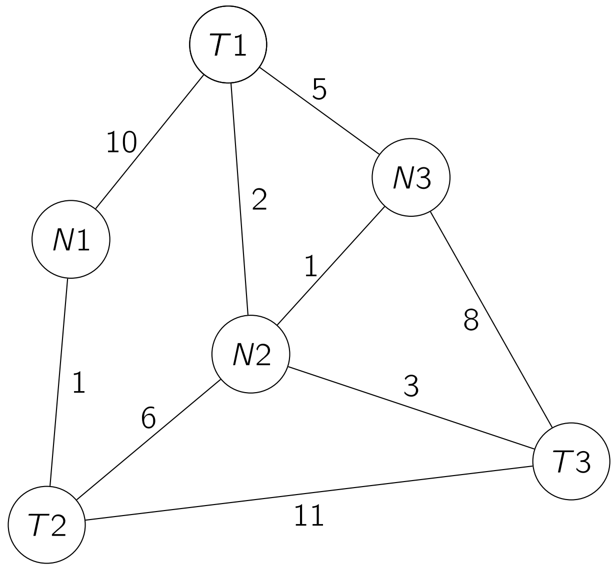
This can be modelled with a weighted undirected graph $G = (V, E)$ with non-negative edge weights. The nodes represent neighborhoods, edges represent the existing cables between the neighborhoods, and edge weights represent the cost to install a new cable.

Because of limited resources and to minimize costs, Algorithmia has chosen to start with the neighborhoods with highest demand for this high speed internet access that will lead to the highest profit, creating a set $T \subset V$ of terminals which includes the neighborhood with Algorithmia's headquarters along with the high demand neighborhoods.

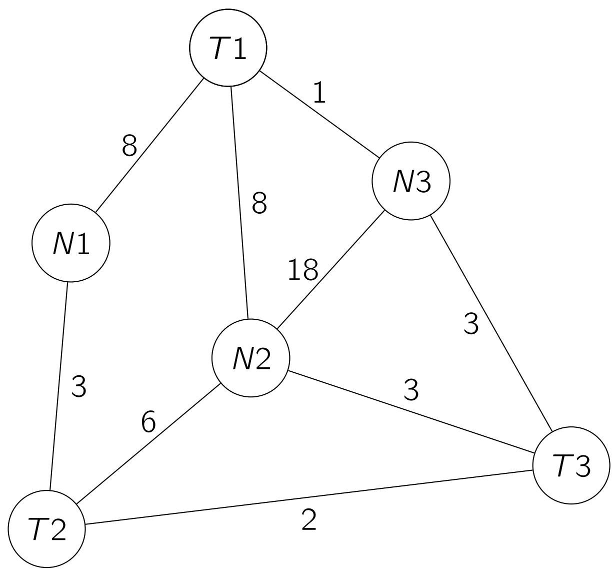
To connect all these neighborhoods, we can model this with a *Steiner tree*, a tree (aka graph with no cycles) that contains all of the terminals and possibly some other vertices. Algorithmia wants to find the minimum weight Steiner tree to find the lowest cost to install new cables that connect Algorithmia's headquarters and the high demand neighborhoods.

(a) If there are only two terminals (Algorithmia's headquarters and another high demand neighborhood), give an $O(n\log(n) + m)$-time algorithm for finding a minimum weight Steiner tree. [**We are expecting:** *English description and brief running time analysis*]

(b) For terminals $T1, T2, T3$, **draw** the minimum weight Steiner tree in each of the following graphs: [**We are expecting:** *A drawing of the Steiner trees.*] (You can use copy+paste the tikz code on tex and delete the lines corresponding to edges that do not participate in the Steiner tree.)



i.



ii.

(c) **Bonus:** Give an $O(n^2 \log(n) + nm)$-time algorithm for finding a minimum weight Steiner tree with **three terminals**. [**We are expecting:** *English description, pseudocode, and running time analysis*]