Graph Problem Solving

Jordi Cortadella and Jordi Petit
Department of Computer Science

Motivation

- Here is a compendium of problems that can be solved using graph algorithms

- No hints are given about the algorithm that must be used, although the statement of the problem makes it evident in many cases

- Strategy:
  - Model the problem as a graph (vertices, edges, weights, etc.)
  - Choose an algorithm to solve a problem on the model
  - Find an interpretation of the solution in terms of the original problem

New road (from [DPV2008])

There is a network of roads $G = (V, E)$ connecting a set of cities V. Each road in E has an associated length l_e. There is a proposal to add one new road to this network, and there is a list E' of pairs of cities between which the new road can be built. Each such potential road $e' \in E'$ has an associated length.

As a designer for the public works department you are asked to determine the road $e' \in E'$ whose addition to the existing network G would result in the maximum decrease in the driving distance between two fixed cities s and t in the network. Give an efficient algorithm for solving this problem.

Streets in Computopia (from [DPV2008])

The police department in the city of Computopia has made all streets one-way. The mayor contends that there is still a way to drive legally from any intersection in the city to any other intersection, but the opposition is not convinced. A computer program is needed to determine whether the mayor is right. However the city elections are coming up soon, and there is just enough time to run a *linear-time* algorithm.

a) Formulate this problem graph-theoretically, and explain why it can indeed be solved in linear time.

b) Suppose it now turns out that the mayor's original claim is false. She next claims something weaker: if you start driving from town hall, navigating one-way streets, then no matter where you reach, there is always a way to drive legally back to the town hall. Formulate this weaker property as a graph-theoretic problem, and carefully show how it too can be checked in linear time.

Blood transfusion

Enthusiastic celebration of a sunny day at a prominent northeastern university has resulted in the arrival at the university's medical clinic of 169 students in need of emergency treatment. Each of the 169 students requires a transfusion of one unit of whole blood. The clinic has supplies of 170 units of whole blood. The number of units of blood available in each of the four major blood groups and the distribution of patients among the groups is summarized below.

Blood type	A	B	O	AB
Supply	46	34	45	45
Demand	39	38	42	50

Type A patients can only receive type A or O; type B patients can receive only type B or O; type O patients can receive only type O; and type AB patients can receive any of the four types.

Give a graph formulation of the problem that determines a distribution that satisfies the demands of a maximum number of patients.

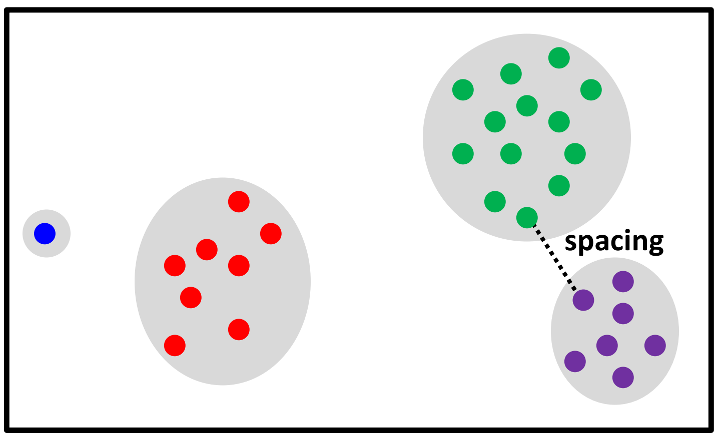
Can we have enough blood units for all the students?

Source: Sedgewick and Wayne, Algorithms, 4th edition, 2011.

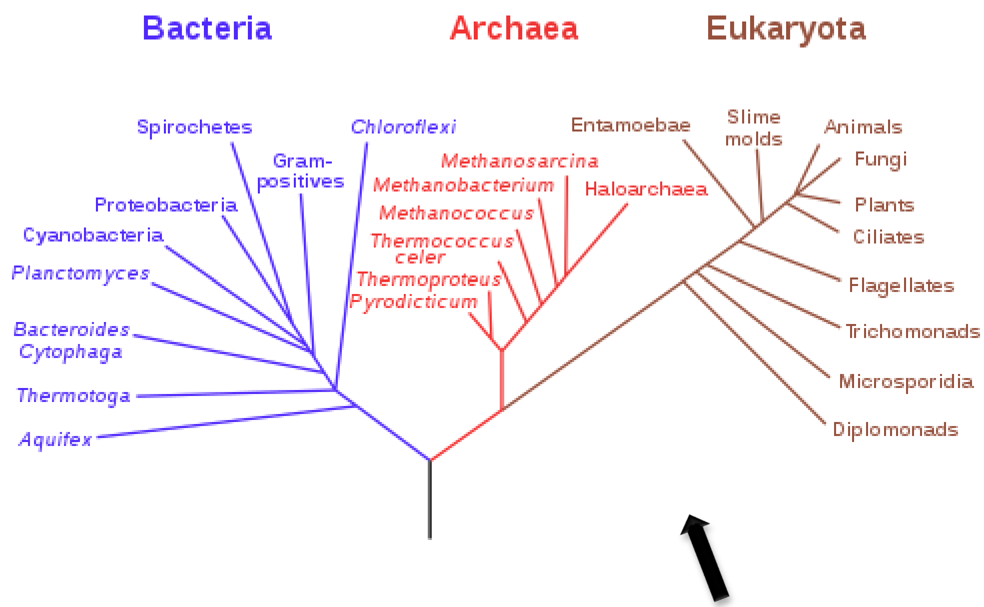
k-clustering of maximum spacing

We want to classify a set of points into k clusters. We define the distance between two points as the Euclidean distance. We define the *spacing* of the clustering as the minimum distance between any pair of points in different clusters.

Describe an algorithm such that, given an integer k, finds a k-clustering such that *spacing* is maximized. Argue about the complexity of the algorithm.



$k = 4$



Note: k-clustering of maximum spacing is the basis for the construction of dendograms.

Nesting boxes

A d-dimensional box with dimensions $(x_1, x_2, \dots, x_d)$ nests within another box with dimensions $(y_1, y_2, \dots, y_d)$ if there exists a permutation π on $\{1,2, \dots, d\}$ such that:

$$x_{\pi(1)} < y_1, x_{\pi(2)} < y_2, \dots, x_{\pi(d)} < y_d.$$

a. Argue that the nesting relation is transitive.

b. Describe an efficient method to determine whether or not one d-dimensional box nests inside another.

c. Suppose that you are given a set of n d-dimensional boxes $\{B_1, B_2, \dots, B_n\}$. Describe an efficient algorithm to determine the longest sequence $\langle B_{i_1}, B_{i_2}, \dots, B_{i_k}\rangle$ of boxes such that B_{i_j} nests within $B_{i_{j+1}}$ for $j = 1,2, \dots, k-1$. Express the running time of your algorithm in terms of n and d.

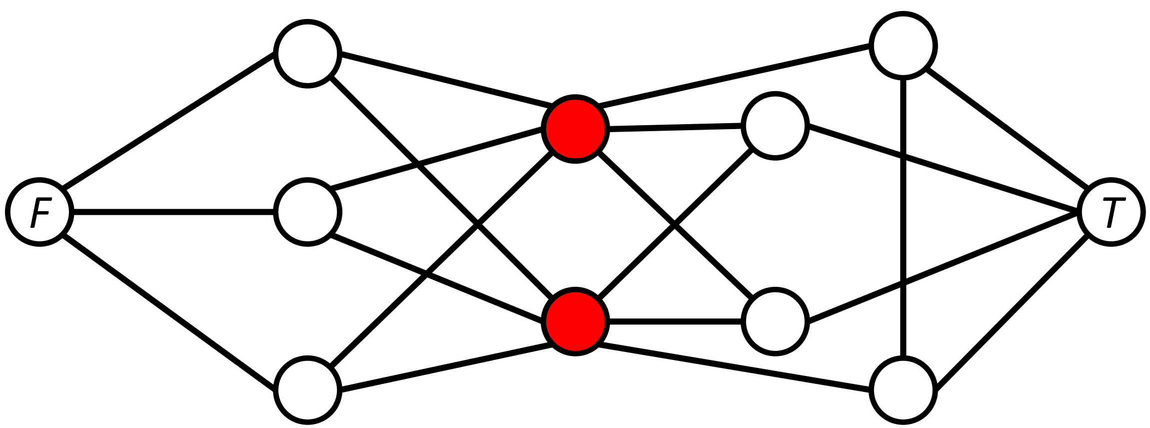
Source: Cormen, Leiserson and Rivest, Introduction to Algorithms, The MIT Press.

Contagious disease

The island of Sodor is home to a large number of towns and villages, connected by an extensive rail network. Recently, several cases of a deadly contagious disease (Covid 19) have been reported in the village of Ffarquhar. The controller of the Sodor railway plans to close down certain railway stations to prevent the disease from spreading to Tidmouth, his home town. No trains can pass through a closed station. To minimize expense (and public notice), he wants to close down as few stations as possible. However, he cannot close the Ffarquhar station, because that would expose him to the disease, and he cannot close the Tidmouth station, because then he couldn't visit his favorite pub.

Describe and analyze an algorithm to find the minimum number of stations that must be closed to block all rail travel from Ffarquhar to Tidmouth. The Sodor rail network is represented by an undirected graph, with a vertex for each station and an edge for each rail connection between two stations. Two special vertices *F* and *T* represent the stations in Ffarquhar and Tidmouth.

For example, given the following input graph, your algorithm should return the number 2.



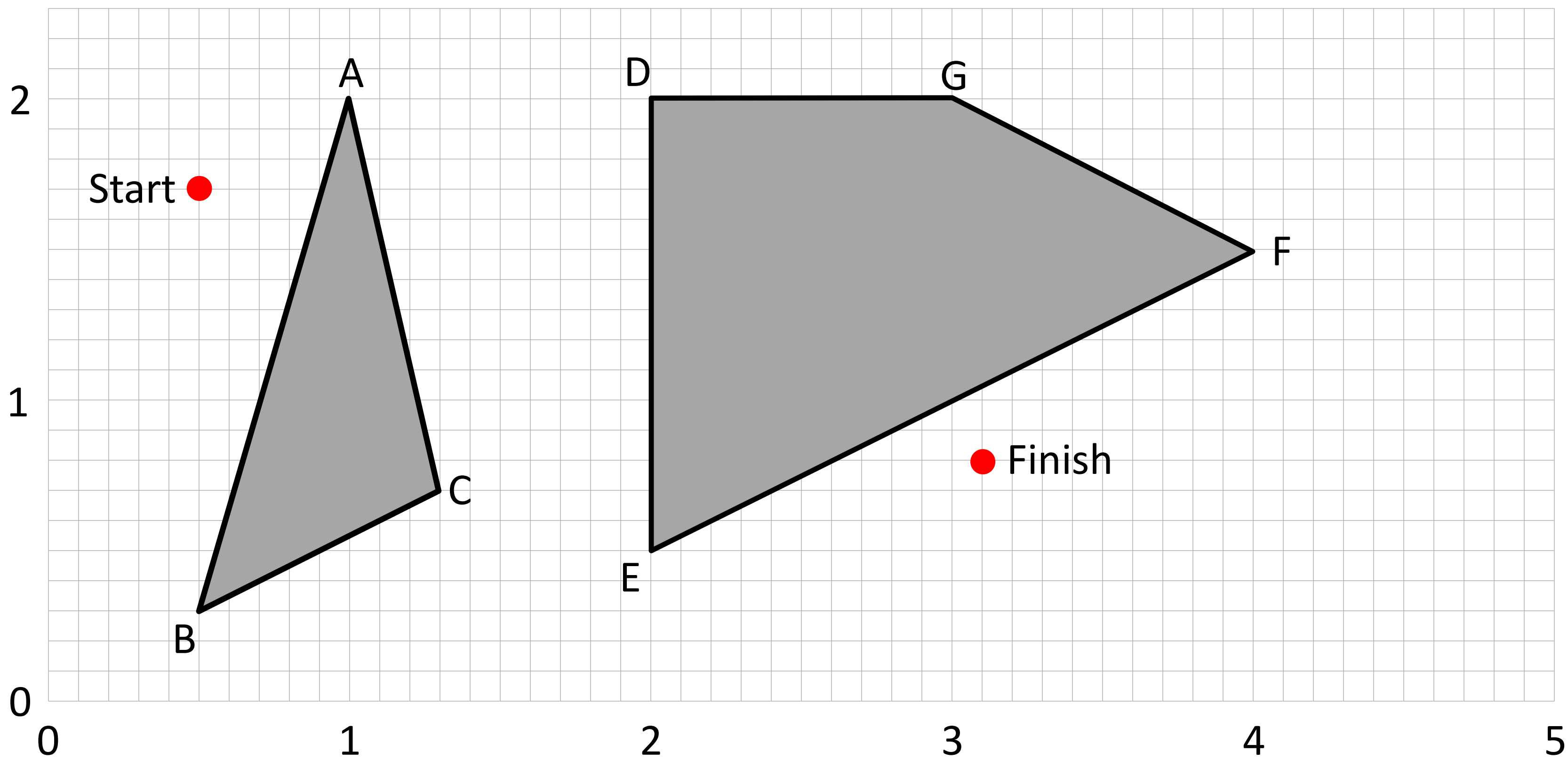
Source: Jeff Erickson, Algorithms, UIUC, 2015.

Edge-disjoint paths

Given a digraph $G = (V, E)$ and vertices $s, t \in V$, describe an algorithm that finds the maximum number of edge-disjoint paths from s to t.

Note: two paths are edge-disjoint if they do not share any edge.

Robot Path Planning



- Find the shortest path from Start to Finish without crossing any gray region
- Show all the steps of the selected algorithm
- Hint: moving to a point in the middle of an open area does not help. The shortest path can be found by visiting vertices of the gray areas

Pouring water (from [DPV2008])

We have three containers whose sizes are 10 pints, 7 pints and 4 pints, respectively. The 7-pint and 4-pint containers start out full of water, but the 10-pint container is initially empty. We are allowed one type of operation: pouring the contents of one container into another, stopping only when the source container is empty or the destination container is full. We want to know if there is a sequence of pourings that leaves exactly 2 pints in the 4-pint container.

a) Model this as a graph problem: give a precise definition of the graph involved and state the specific question about this graph that needs to be answered.

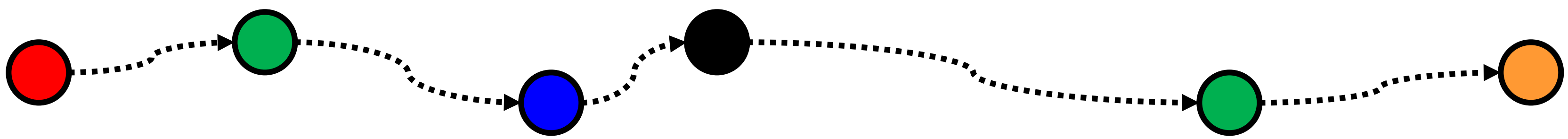
b) What algorithm should be applied to solve the problem?

c) Give a sequence of pourings, if it exists, or prove that it does not exist any sequence.

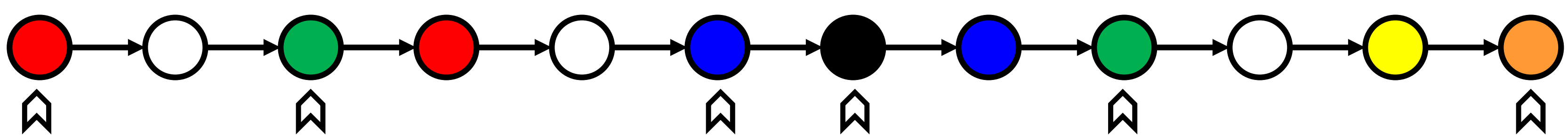
Hint: A vertex of the graph can be represented by a triple of integers.

Colored path

- Let us consider a directed graph in which each node is assigned a color and each edge is assigned a positive distance. Let us also consider a sequence of k colors, e.g.,



- Problem: find the shortest path that visits nodes with the given sequence of colors (and possibly other nodes in the middle), e.g., for the previous sequence, a valid path could be:



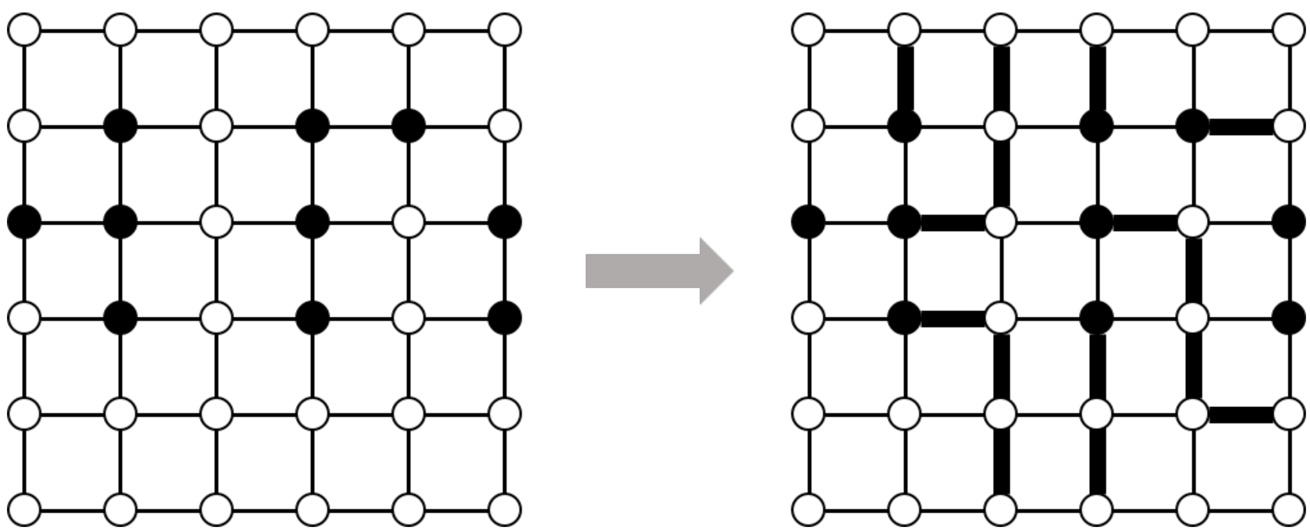
- Discuss the computational complexity of your solution.

The escape problem [CLR]

An $n \times m$ grid is an undirected graph with $n \times m$ vertices organized in n rows and m columns. We denote the vertex in the i-th row and the j-th column by (i, j). Every vertex (i, j) has exactly four neighbors $(i-1, j)$, $(i+1, j)$, $(i, j-1)$ and $(i, j+1)$, except the boundary vertices, that have fewer neighbors.

Let (x_1, y_1), (x_2, y_2), …, (x_k, y_k) be distinct vertices, called terminals, in the $n \times m$ grid. The escape problem is to determine whether there are k vertex-disjoint paths in the grid that connect the terminals to any k distinct boundary vertices.

Hint: maxflow (the graph model is not obvious)



Critical edge

Let us consider a strongly connected graph $G(V, E)$ with positive weights $w_e > 0$ on the edges. We would like to answer the following question: given $s, t \in V$, and $e \in E$, do all shortest paths from s to t contain e?

Find an efficient algorithm to answer this question and discuss the complexity of the algorithm.

You do not need to give the code of the algorithm. It is sufficient if you describe a precise strategy to solve the problem.