Lecture At the lecture we will talk about network flows. We will talk about maximum flows, their relation to minimum cuts, and residual networks. You should read CLRS chapter 26.1-26.2.

Exercises

Fibonacci numbers Solve CLRS 15.1-5.

Planning a company party Solve CLRS 15-6.

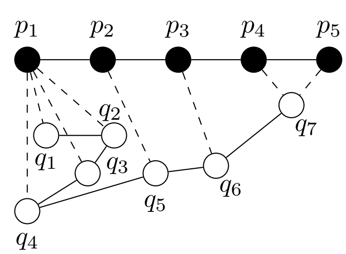
All pairs shortest paths Solve CLRS 25.1-1 (only Faster-APSP), 25.1-9, 25.1-10.

Discrete Fréchet distance Consider Professor Bille going for a walk with his personal dog. The professor follows a path of points $p_1, \ldots, p_n$ and the dog follows a path of points $q_1, \ldots, q_m$. We assume that the walk is partitioned into a number of small steps, where the professor and the dog in each step either both move from p_i tp p_{i+1} and from q_j to q_{j+1}, respectively, or only one of them moves and the other one stays.

The goal is to find the smallest possible length L of the leash, such that the professor and the dog can move from p_1 and q_1, resp., to p_n and q_n. They cannot move backwards, and we only consider the distance between points. The distance L is also known as the discrete Fréchet distance.

We let $L(i, j)$ denote the smallest possible length of the leash, such that the professor and the dog can move from p_1 and q_1 to p_i and q_j, resp. For two points p and q, let $d(p, q)$ denote the distance between the them.

In the example below the dotted lines denote where Professor Bille (black nodes) and the dog (white nodes) are at time 1 to 8. The minimum leash length is $L = d(p_1, q_4)$.



Q1: Recurrence Give a recursive formula for $L(i, j)$.

Q2: Algorithm Give pseudo code for an algorithm that computes the length of the shortest possible leash. Analyze space and time usage of your solution.

Q3: Print solution Extend your algorithm to print out paths for the professor and the dog. The algorithm must return where the professor and the dog is at each time step. Analyze the time and space usage of your solution.

Puzzle of the week: 100 Ants [1]There are 100 ants on a rod of length 1 metre. The ants are arbitrarily positioned on the rod and are travelling at 1 metre/minute either right or left at the start. The ants are also perfectly elastic, so that if two ants collide they simply turn round and carry on at 1 metre/minute in the opposite direction. If an ant reaches the end of the rod it falls off. The question is, what is the longest time it can take for all 100 ants to fall off the rod?

You can assume the ants are points on the rod and that the rod is simply a one dimensional line (i.e. ants can only go left or right). You are asked to find the longest time it can take for all the ants to fall off the rod over all possible start states and starting directions of the ants.

[1]I got this puzzle from Raphäel Clifford

Mandatory exercise: "Bing or Ding" (from the exam E15) At the Christmas party there will be a game called "Bing or Ding". There is a row of n red socks, each with an integer on it, which can be positive, negative, or zero. Let s_i denote the number on sock i. The rules are as follows:

- You have to consider all socks in order from sock 1 to n.
- At each sock you *must* say either "Bing!" or "Ding!".
- If you say "Bing!" at the ith sock you get s_i Christmas cookies.
- If you say "Ding!" at the ith sock you must pay s_i Christmas cookies.
- It is possible to end up with a negative number of Christmas cookies.
- It is forbidden to say the same word more than two times in a row. For example if you say "Bing!" at sock 5 and 6 you *must* say "Ding!" at sock 7.

Let $L(1, i)$ denote the maximum number of Christmas cookies you can have after the first i socks if you say "Bing!" at sock i. Similarly, let $L(0, i)$ denote the maximum number of Christmas cookies you can have after the first i if you say "Ding!" at sock i. Let $L(b, 0) = 0$ for $b \in \{0, 1\}$.

Question 1 Fill out the table below for $L(b, i)$ when $S = [s_1, s_2, s_3, s_4] = [3, 4, 8, -4]$.

$L(b,i)$	0	1	2	3	4
0					
1					

Question 2 Which of the following recurrences correctly computes $L(b, i)$:

A
$$L(b,i) = \begin{cases} 0 & \text{if } i = 0 \\ L(0, i-1) + s_i & \text{if } b = 1 \\ L(1, i-1) - s_i & \text{if } b = 0 \end{cases}$$

B
$$L(b,i) = \begin{cases} 0 & \text{if } i = 0 \\ s_1 & \text{if } b = 1 \text{ and } i = 1 \\ -s_1 & \text{if } b = 0 \text{ and } i = 1 \\ \max\{L(1, i-2) - s_{i-1}, L(0, i-2) + s_{i-1}\} + s_i & \text{if } b = 1 \text{ and } i \geq 2 \\ \max\{L(1, i-2) + s_{i-1}, L(0, i-2) - s_{i-1}\} - s_i & \text{if } b = 0 \text{ and } i \geq 2 \end{cases}$$

C
$$L(b,i) = \begin{cases} 0 & \text{if } i = 0 \\ s_1 & \text{if } b = 1 \text{ and } i = 1 \\ -s_1 & \text{if } b = 0 \text{ and } i = 1 \\ \max\{L(0, i-1), L(0, i-2) + s_{i-1}\} + s_i & \text{if } b = 1 \text{ and } i \geq 2 \\ \max\{L(1, i-1), L(1, i-2) - s_{i-1}\} - s_i & \text{if } b = 0 \text{ and } i \geq 2 \end{cases}$$

Question 3 Write pseudocode for an algorithm *based on dynamic programming and the recurrence from Question 2* that finds the maximum number of Christmas cookies you can end up with given n socks. Analyze the space usage and running time of your algorithm in terms of n.