

## 1 Asymptotics

- (a) Prove or disprove:  $n! = \Omega(2^n)$ .
- (b) Prove or disprove:  $n^{\log n} = O((\log n)^n)$ .

## 2 Recurrences

Motivate all answers. Explicitly state methods used.

- (a) Solve exactly:  $T(1) = 3$ , for all  $n \geq 2$ ,  $T(n) = T(n-1) + 2n - 3$ .
- (b) Give asymptotically tight solution:  $T(1) = 1$ , for all  $n \geq 2$ ,  $T(n) = 4T(\frac{2n}{3}) + n^3$ .
- (c) Solve the recurrence  $T(1) = 1$ , for all  $n \geq 2$ ,  $T(n) = 2T(\sqrt{n}) + 1$  by making a change of variables. Your solution should be asymptotically tight. Do not worry about whether values are integral.

## 3 Search in an Array

Let  $T[1..n]$  be a sorted array of distinct integers. Give a divide-and-conquer algorithm that finds an index  $i$  such that  $T[i] = i$ , if one exists. Analyze its running time.

**Hint:** More efficient algorithms always give you better scores.

## 4 Creek Crossing in the Dark

- (a) It's pitch dark, but a team of four hikers need to cross a narrow shaky bridge that can carry at most two of them at a time. They have just one flashlight necessary to cross. Knowing that they need 1, 2, 5, and 10 minutes, respectively, to cross the bridge in one direction, suggest an optimal schedule allowing the team to get to the opposite side in the *shortest* possible time.
- (b) Generalize: there are  $n$  people, each needs  $t_i$  minutes to cross. Design the algorithm determining the optimal time and schedule. Explain and justify heuristics used, and analyze complexity of your algorithm.

## 5 Karlsson-on-the-Roof Strikes Again (Karlsson på Taket Smyger Igen)

Lillebror wants to move his truck from city A to city B as cheaply as possible. He works hard driving the whole day from city  $x$  to an adjacent city  $y$ . While he sleeps at night in the cabin of his truck, Karlsson who lives on the roof of the truck, drives from  $y$  to an adjacent city  $z$ , with the aim to maximize the overall trip cost.

Assuming the map is given as a connected, directed *acyclic* weighted graph (edge weights indicate costs of the trip between adjacent cities) where the only sink<sup>1</sup> is B, suggest optimal algorithms for Lillebror and Karlsson, allowing both to achieve their goals, and computing how much Lillebror has to shell out, if both act optimally. Analyze efficiency of your algorithms and explain techniques used. See Figure 1 for an example input.

**Hint:** You might want to start by considering restricted bipartite graphs and extend afterwards. An example of the map where Lillebror starts driving and both can secure the cost 1200 is given in Figure 1.

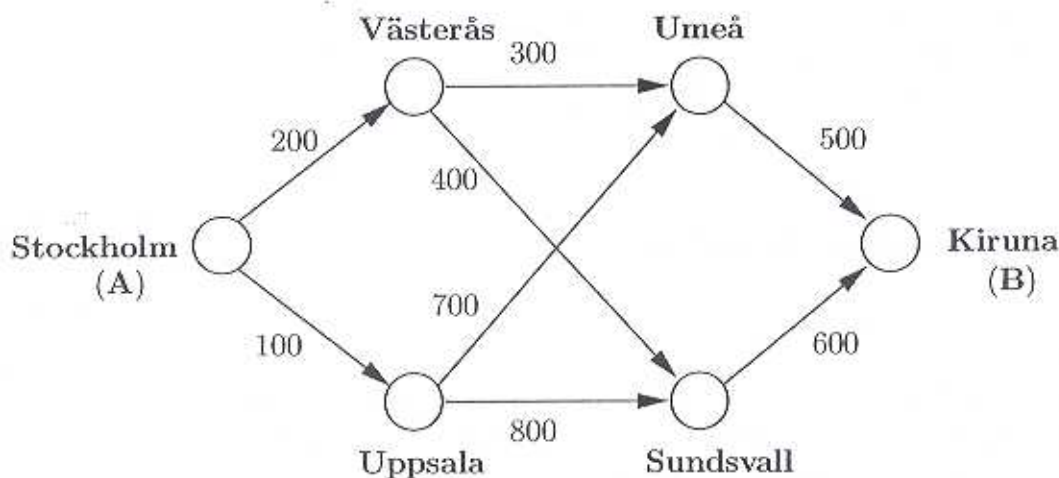


Figure 1. Lillebror wants to go from A to B. He can be sure of paying  $\leq 1200$ , while Karlsson can secure the cost  $\geq 1200$ .

## 6 Disjoint Sets

Show the data structure that results (give snapshots before lines (5) and (7), as well after lines (7), (8), (9)) and the answers returned by the Find-Set operations in the following program. Use the disjoint set forest representation with union by size and path compression.

**Hint:** When making a UNION of two equal size sets (trees) make the root of the second one the parent of the first; otherwise the bigger becomes the parent of the smaller.

**Algorithm 1:** Sequence of Disjoint Set Operations

- (1) for  $i \leftarrow 1$  to 16
- (2) MAKE-SET( $x_i$ )
- (3) for  $i \leftarrow 1$  to 15 by 2
- (4) UNION( $x_i, x_{i+1}$ )
- (5) for  $i \leftarrow 1$  to 13 by 4
- (6) UNION( $x_i, x_{i+2}$ )
- (7) UNION( $x_1, x_5$ )
- (8) UNION( $x_{11}, x_{13}$ )
- (9) UNION( $x_1, x_{10}$ )
- (10) FIND-SET( $x_2$ )
- (11) FIND-SET( $x_9$ )

## 7 Penguin Post Problem

In Penguinville, Antarctica, there is no regular post system. Penguins don't have secrets, so instead they use the following system to deliver messages. To send a message to another penguin, you first give it to all your penguin friends in the morning. The next morning, your friends will give it to their friends, and eventually the message will spread to every penguin that is a friend of a friend of a friend... So it takes three days to deliver a message to a friend of a friend of a friend. It is known which penguins are friends and which are not. If one penguin is a friend of another, then the other is also a friend of the first. The penguins now want to know what the longest time to send a message is (and if every message reaches every penguin). Invent an algorithm that solves this problem and analyze running time.

## 8 Living on the Edge

Stora Karlsö, a hat-shaped island west of Gotland, is famous for its population of Guillemots<sup>2</sup>, a type of bird that nest on the steep cliff sides of the island. The cliffs are grid-shaped like in Figure 2, and each slot may be inhabited by a Guillemot family. Two Guillemot families are *neighbors* if they share a wall: either the wall above, below, to the left or to the right. A *colony* is a non-empty group of neighboring families. So there are four colonies in Figure 2: one with four members, one with three, one with seven and one with six. Give an algorithm that helps the Guillemots to count the number of colonies and the number of members in each colony. The input is represented as an  $n \times m$ -matrix with 0 in empty places and 1 in inhabited places.

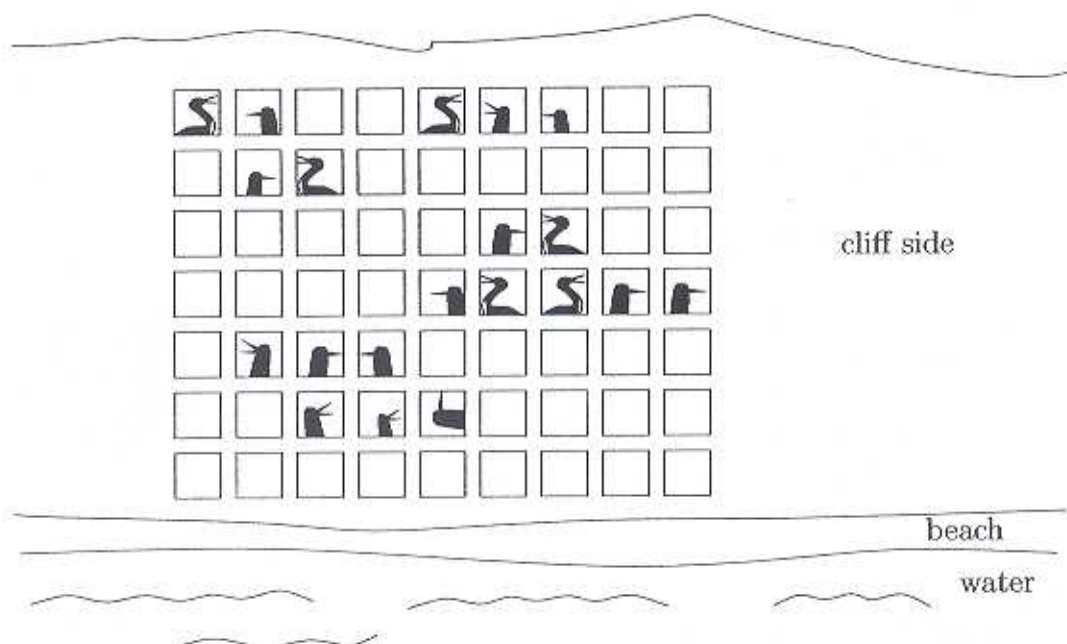


Figure 2. Guillemots living on a cliff side of Stora Karlsö

## 9 Pucks Shipping

Show the execution of the Edmonds-Karp algorithm on the flow network in Figure 3, find the maximum flow, and show the corresponding minimum cut.

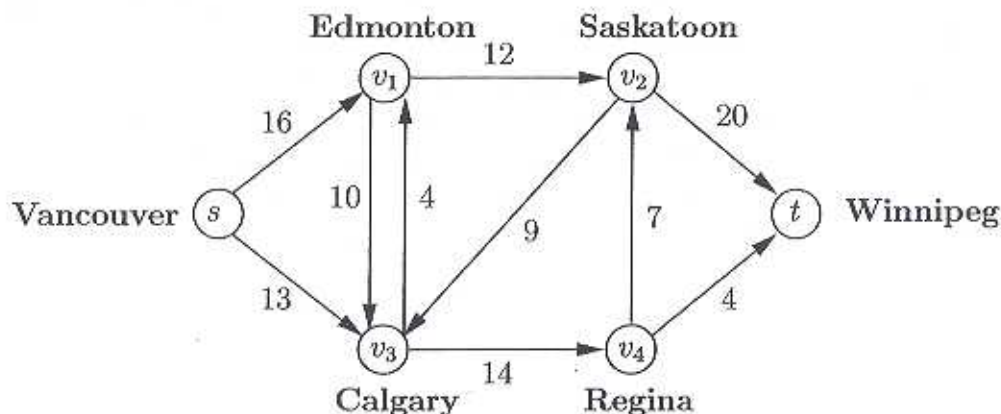


Figure 3. A flow network for the Lucky Puck Company's trucking problem.

## 10 Printing All Segment Intersections

Professor Baltazar suggests that we modify ANY-SEGMENTS-INTERSECT so that instead of returning upon finding an intersection, it prints the segments that intersect and continues on to the next iteration of the for loop. The professor calls the resulting procedure PRINT-INTERSECTING-SEGMENTS and claims that it prints all intersections, from left to right, as they occur in the set of line segments. Show both claims are wrong.

**Hint:** Give a set of segments for which the first intersection found by PRINT-INTERSECTING-SEGMENTS is not the leftmost one and a set for which PRINT-INTERSECTING-SEGMENTS fails to find all the intersections.