Mathematical Foundations of Computer Science Module: CH-233
Constructor University Date: 2025-09-05
Dr. Jürgen Schönwälder Due: 2025-09-12

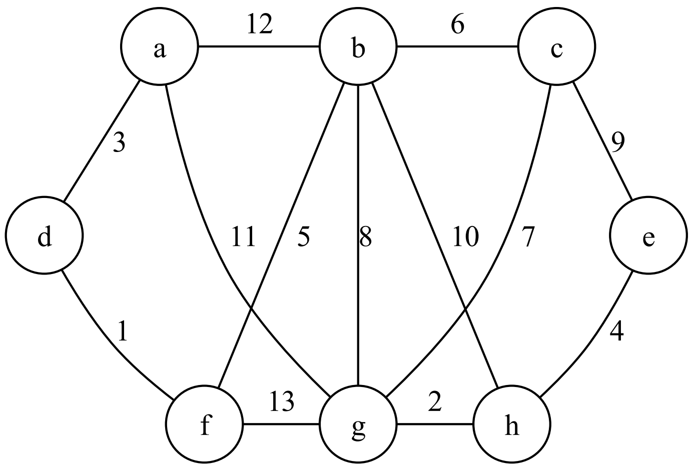
Problem Sheet #1

Problem 1.1: *minimum spanning trees* (2+3 = 5 points)

We have introduced Kruskal's algorithm for constructing random spanning trees. Edges were selected randomly and added to the spanning tree as long as the nodes connected by the edges belong to different equivalence classes.

Kruskal's algorithm solves a slightly more difficult problem: Given a graph $G = (V, A)$ and a cost function $c\colon A \to \mathbb{R}$ indicating the cost of the edges, calculate the spanning tree $T = (V, E)$ with $E \subseteq A$ such that $s = \sum_{e \in E} c(e)$ is minimal (also called a minimum spanning tree). Kruskal's algorithm solves this problem by selecting in each step an edge that joins two equivalence classes and has the minimum cost of all edges still available.

Consider the following graph. The labels next to the edges indicate the cost of the different edges.



a) Define the graph $G = (V, A)$ shown above by defining the set of verticies V and the set of edges A (ignore the costs here).

b) Construct a minimal spanning tree $T = (V, E)$ using Kruskal's algorithm. For each step, write down the set of equivalence classes C and the edges in E. What is the overall cost S of the resulting spanning tree? You start with:

$E = \{\}$ initialization, $s = 0$
$C = \{\ldots\}$

$E = \ldots$ step 1, $s = \ldots$
$C = \ldots$

$E = \ldots$ step 2, $s = \ldots$
$C = \ldots$

$\ldots$

Problem 1.2: *boyer moore algorithm* (1+2+2 = 5 points)

You have implemented a run and jump game, which is controlled by a game controller sending a sequence of run (R), jump (J), wait (W), and turn (T) control signals. A game play thus can be expressed with a sequence such as RRJRJRWRWRRJRWR. Using the Boyer Moore algorithm, we can determine whether a sequence of a game play contains certain control sequences.

Let $\Sigma = \{J, R, T, W\}$ be an alphabet and $t \in \Sigma^*$ be a text of length n describing a program for the robot. Let $p \in \Sigma^*$ be a pattern of length m. We are looking for the first occurrence of p in t.

Consider the text $t = RRJRJRWRWRRJRWR$ and the pattern $p = RRJRW$.

a) Execute the naive string search algorithm. Show all alignments and indicate comparisons performed by writing uppercase characters and comparisons skipped by writing lowercase characters. How many alignments are used? How many comparisons are done?

b) Execute the Boyer-Moore string search algorithm with the bad character rule only. How many alignments are used? How many comparisons are done?

c) Calculate the lookup table for the bad character rule that indicates the number of alignments that can be skipped if a comparison does not match.