Marburg University

Department of Computer Science
Prof. Dr. Sebastian Wild

Date: 2025-01-16
Version: 2026-01-16 17:52

Exercise Sheet 11 for Effiziente Algorithmen (Winter 2025/26)

Hand In: *Until 2026-01-23 18:00, on ILIAS.*

Problem 1

20 + 10 + 30 points

Given an array $B[0 \dots n)$ with n boolean values (n bits). In the following, a *logical AND* ($\wedge$) is to be computed on the array. The result is *True* exactly when all n entries in the array are *True*. (We assume that each bit is stored as a whole word.)

a) Design a CREW-PRAM parallel algorithm that computes the *logical AND* on $B[0 \dots n)$. The algorithm should have a time (span) of $\mathcal{O}(\log n)$ and a work of $\mathcal{O}(n \log n)$.

b) Can you make the algorithm work-efficient?

c) Now consider the CR**C**W-PRAM model. You may choose a conflict strategy that you deem appropriate. Design a parallel algorithm that computes the *logical AND* in *constant* time.

Problem 2

30 + 30 points

In the knapsack problem, a set of n objects and a weight limit W are given.

Each object i has a value v_i and a weight w_i. The problem is to select a subset S of the n objects such that the total value $\sum_{i \in S} v_i$ is maximized under the constraint $\sum_{i \in S} w_i \le W$. We assume in the following that all value and weight numbers are non-negative real numbers.

The fractional knapsack problem is a variant where each object i can be packed into the knapsack at any fraction $0 < b \le 1$ (thus with weight $b \cdot w_i$ and value $b \cdot v_i$).

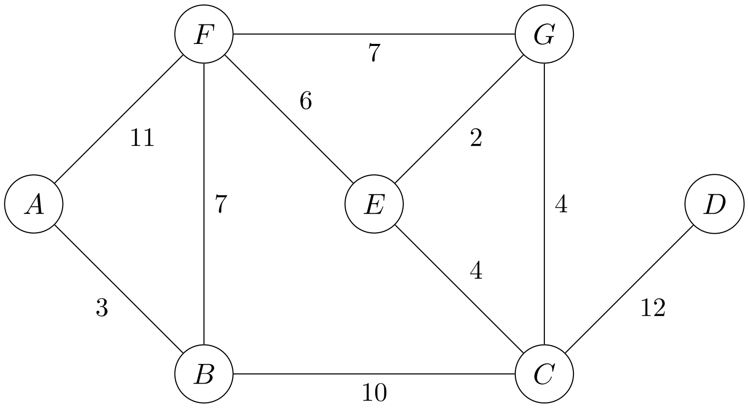
a) Design a greedy algorithm to compute a solution for the simple (0/1) variant as well as for the fractional variant of the knapsack problem. The solution for the fractional variant should be optimal.

b) Show that the greedy method for the 0/1 knapsack can become arbitrarily bad. Argue why the solution for the fractional knapsack guarantees an optimal solution.

Problem 3

20 + 20 points

a) Compute a minimum spanning tree for the following graph using Kruskal's algorithm. Also provide all other possible minimum spanning trees.



b) Show: For a graph G and a minimum spanning tree T, the input for Kruskal's algorithm can be adjusted so that Kruskal's algorithm yields T as the result.

Problem 4

40 points

Professor Caesar has designed a new divide-and-conquer algorithm for computing minimum spanning trees.

For a graph $G = (V, E)$ the set of vertices V is divided into two sets V_1 and V_2 such that $||V_1| - |V_2|| \leq 1$. Let E_1 be the set of edges that are incident only to vertices in V_1 and E_2 the set of edges that are incident only to vertices in V_2. The problem is solved recursively on the two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Afterwards, one selects the edge in E with minimum weight that crosses the cut (V_1, V_2) in order to connect the two minimum spanning trees into a new spanning tree.

Show or refute: The algorithm computes a minimum spanning tree for G.