

# Algorithms SV worksheet 5

Bálint Molnár

September 28, 2025

*I understand that the order in which the topics are taught has changed slightly from previous years. Before attempting the questions, make sure you know the following concepts/definitions: DAG, Minimum Spanning Tree/Kruskal's Algorithm, Max-flow problem/Ford-Fulkerson algorithm, Topological sort, Strongly Connected Components, Bipartite Graph Matching. You may want to use the [notes from 2024](#).*

1. If we take a DAG and reverse all the edges, do we get a DAG? Briefly justify your answer.
2. Let  $G$  be a weighted graph with positive weights. Show that if you replace all the weights with their squares, the resulting graph has the same Minimum Spanning Tree.
3. You are given an  $N \times N$  grid represented by a matrix  $A$  consisting of 0s and 1s. You want to choose  $N$  cells in such a way that
  - You choose exactly one cell from every row.
  - You choose exactly one cell from every column.
  - You can only choose cells where the corresponding entry is 1.

How would you find such a set of cells?

4. You are given a DAG  $G = (V, E)$ , and a list of special nodes,  $[a_1, \dots, a_k]$ . It is guaranteed that, for all  $1 \leq i < k$ , there is a path from  $a_i$  to  $a_{i+1}$ .

Design an algorithm that finds a topological sort of  $G$  which minimises the ranks for all  $a_i$  in the resulting list.

5. Explain how to solve [this coding problem](#) efficiently. Please include the full code in your preferred programming language, along with an explanation of your approach.
6. You have an island represented by an  $N \times M$  grid, with two harbours located on the boundary (either in the first or last row or first or last column). You want to build a border checkpoint line, a set of cells such that any path between the two harbours must pass through at least one checkpoint cell. A path is a sequence of moves between neighbouring cells (left, right, up or down, no diagonal moves). Checkpoints cannot be built on harbour cells. Each cell  $(i, j)$  has an associated cost  $A[i, j]$  to build a checkpoint.

Find a set of checkpoints that satisfies the requirements at a minimal cost.

**Example**

