TU Wien
Institute of Logic and Computation
Algorithms and Complexity Group

186.814 Algorithmics VU 6.0

Exam (winter term 2020/21)

February 2, 2021

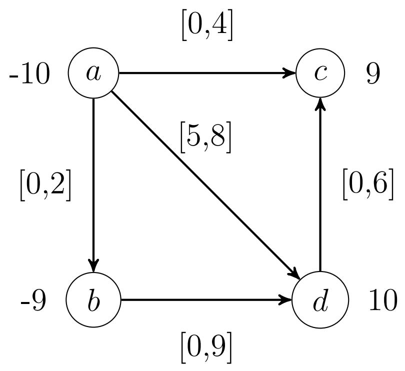
Write your answers on blank paper (with your name and matriculation number on the top of each page), which you finally scan/take pictures of and submit as one PDF file via TUWEL.

For further details we refer to the earlier distributed organizational instructions.

Good luck!

Question 1: (Network Flows) **(5 Points)**

You are given the following circulation problem with supplies, demands, and lower bounds $N = (V, E, s, t, c, d, \ell)$. Each edge e is labeled with a pair $[c(e), \ell(e)]$ where $c(e)$ is its capacity and $\ell(e)$ is its lower bound.



a) Formulate an equivalent circulation problem without lower bounds, draw the circulation network N' and indicate capacities and the supplies/demands.

b) Formulate the flow problem which corresponds to a); draw the flow network N'' and label the edges with the respective capacities.

c) Indicate in N'' a maximum flow; you do not need to indicate how you found the flow.

d) Translate the maximum flow for N'' you found in c) into a feasible circulation for the original circulation network N. Specify the circulation in the form $f(a, b) =$, $f(a, c) =$, $f(a, d) =$, $f(b, d) =$, $f(d, c) =$.

Question 2: Fixed Parameter Tractability **(10 Points)**

a) (7 Points)

Consider the following parameterized problem. Recall that a vertex cover is a set X of vertices such that each edge in G has at least one endpoint in X.

RED-BLUE-VERTEX COVER
Instance: A graph G with each vertex colored either *blue* or *red*, and two integers r and b. *Parameter*: $r + b$. *Question*: Does G admit a vertex cover consisting of at most r red vertices and at most b blue vertices?

Show that RED-BLUE-VERTEX COVER has a polynomial kernel by describing a kernelization algorithm for the problem. In particular, (1) specify the reduction rules you use, (2) briefly argue that each rule is safe, and (3) explain why these rules result in a polynomial kernel. *Hint: Recall the polynomial kernelization for* VERTEX COVER *as an inspiration.*

b) (3 Points)

For each of the following properties, either give an example of a parameterized problem with these properties, or correctly identify that no such parameterized problem exists (assuming standard complexity-theoretic assumptions). If a problem with the listed properties exists, please describe it using the following format:

Instance: *Parameter*: *Question*:

- A parameterized problem which is neither FPT nor XP.
- A decidable parameterized problem which admits a kernel but is not FPT.
- A parameterized problem which is in FPT and also in XP

Question 3: Randomized Algorithms **(10 Points)**

Assume that you have a biased coin that when flipped yields head with some *unknown* probability p with $1/2 < p < 1$.

a) (3 points) How can you use this coin to create a string of n independent unbiased random bits? State the algorithm in simple pseudo-code.
Hint: Consider pairs of coin flips.

b) (2 points)

Formally argue with concrete probabilities that indeed each bit returned by your algorithm is equally likely 0 or 1.

c) (2 points) What is the expected number of coin flips of your algorithm in dependence of n and p? Argue it.

d) (3 points) Is your algorithm a Monte Carlo or Las Vegas Algorithm? Argue why. Moreover, provide an example how you can modify your algorithm in order to turn it into the other kind of randomized algorithms.

Question 4: Structural Decompositions and Algorithms (5 Points)

An *induced path* of a graph G is a path that is an induced subgraph of G. That is, no two non-consecutive vertices on the path are adjacent. The LONGEST INDUCED PATH problem asks for a maximum-length induced path of a given input graph G (note that this is an optimization problem). Use FPT results for MSO model checking to prove that this problem is FPT parameterized by the treewidth of the graph G.

You may use (without defining) a formula $\mathit{connected}(X)$ such that $G \models \mathit{connected}[V]$ if, and only if, V is a subset of vertices that is connected in G.

Question 5: Linear Programming **(10 Points)**

a) (3 Points)

Consider the following primal LP:

$$\begin{aligned} \min \quad & 2x_1 + 5x_2 && (1) \\ & -4x_1 + 7x_2 \leq 6 && (2) \\ & 9x_1 - x_2 \geq 8 && (3) \\ & x_1 \in \mathbb{R} && (4) \\ & x_2 \geq 0 && (5) \end{aligned}$$

Form the corresponding dual LP.

For the following subtasks, consider the problem: Given is an undirected simple graph $G = (V, E, c)$ with nodes $V = \{1, \dots, n\}$ and cost function $c\colon E \to \mathbb{R}^+$. We call node 1 the *source*. Furthermore let $S \subseteq V \setminus \{1\}$ be a subset of nodes that we call *sinks*. Our aim is to find a tree being a connected acyclic subgraph of G that connects all the sinks with the source node 1 such that the sum of the edge costs is minimal.

We want to model this problem as a MILP and define the following two sets of variables:

$$\begin{aligned} x_{ij} &\in \{0, 1\} \quad \forall (i, j) \in E && (6) \\ y_j &\in \{0, 1\} \quad \forall j \in V && (7) \end{aligned}$$

If $x_{ij} = 1$, then edge $(i, j) \in E$ is used, and $y_j = 1$ indicates that node j is connected in the resulting tree. Note that the resulting tree usually does not contain every node.

b) (1 Point)

Formulate the objective function.

c) (1.5 Points)

Link the y_j variables correctly with the x_{ij} variables: A node is part of the resulting tree, if at least one incident edge is chosen.

d) (3 Points)

Find a correct flow formulation to ensure that the resulting solution forms an acyclic tree. In particular: How must the flows be linked to the y_j and x_{ij} variables?

To ease modeling you may also use the directed version of the graph $G' = (V, A)$ with $A = \{(i, j), (j, i) \mid \{i, j\} \in E\}$.

e) (1.5 Points)

Make sure that every sink node is connected to the resulting tree.

Question 6: Geometric Algorithms **(10 Points)**

a) (5 Points)

Consider a set P of n points in $\mathbb{R}^2$ and let a, b, c, d be four distinct points in P, no three of them collinear. Prove the following claim using the properties of Voronoi diagrams that you know from the lecture:

> If neither the circle C through a, b, c nor the circle C' through b, c, d contain any fourth point of P inside them or on the respective circle boundary, then the line segment $\overline{pp'}$ between the centers p of C and p' of C' is exactly the Voronoi edge separating the cells $\mathcal{V}(b)$ and $\mathcal{V}(c)$.

b) (5 Points)

Let $G = (V, E)$ be a simple, connected, planar embedded graph stored as a doubly-connected edge list (DCEL) and let $v \in V$ be a vertex. Give an algorithm in pseudocode (or sufficiently precise description) that returns the set of all outgoing half-edges $e = (v, u)$, which are incident to v and separate two triangular faces of G. The time complexity should be $O(\deg(v))$.

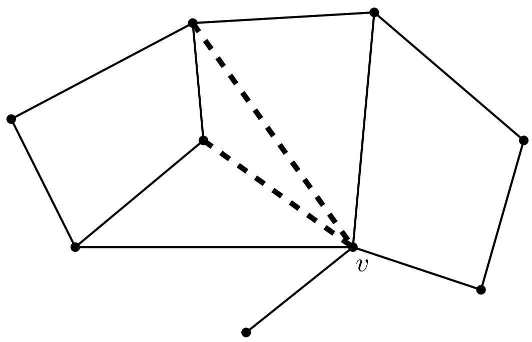


Figure 1: In this example, you should return exactly the two dashed edges (or their respective outgoing half-edges of v).

You can make use of the basic functions of DCEL's defined in the lecture (`edge(v)` for a vertex v, `edge(f)` for a face f, and `origin(e)`, `twin(e)`, `next(e)`, `prev(e)`, `face(e)` for an edge e).