TU Wien
Institute of Logic and Computation
Algorithms and Complexity Group

186.814 Algorithmics VU 6.0

2nd Exam (winter term 2020/21)

March 15, 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.

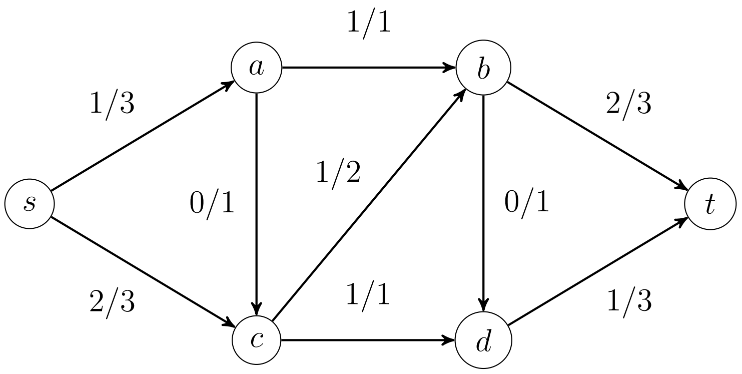
Question	Points	(max)	Question	Points	(max)
1		(10)	4		(10)
2		(10)	5		(5)
3		(5)	6		(10)

Total: (50)

Good luck!

Question 1: Network Flows **(10 Points)**

Consider the following flow network $N = (V, E, s, t, c)$ with flow f, where each edge e is labeled by the pair $f(e)/c(e)$.



a) (1 point) Determine the value of flow f.

b) (2 points) Draw the residual graph G_f.

c) (2 points) Determine the *shortest* augmenting path P in G_f.

d) (2 points) Obtain a new flow f' by augmenting f along the augmenting path P (f' is a maximum flow).

e) (2 points) Draw the residual graph $G_{f'}$.

f) (1 point) Determine the minimum cut (A, B) of N which corresponds to the maximum flow f', by giving the sets A and B.

Question 2: Parameterized Algorithms (10 Points)

Let K_4 be a *clique* on 4 vertices, i.e., the graph consisting of 4 vertices with edges between each pair of vertices. Consider the following problem:

4-Clique Removal
Instance: A graph G and an integer k.
Parameter: k.
Question: Does there exist a set X of at most k vertices in G such that deleting X from G results in a graph with no K_4 as a subgraph?

a) (7 points)

Design and describe a fixed-parameter algorithm for 4-Clique Removal. You are allowed to either give a sufficiently detailed description of the algorithm or provide a high-level pseudocode. You may assume that checking whether a set of 4 vertices forms a clique or not can be performed by a subroutine that runs in constant time (e.g., `isClique`(a, b, c, d)).

Hint: Try using the bounded search tree technique.

Additionally, give an explicit upper bound on the runtime of your algorithm.

b) (3 points)

Recall the Sat and MaxSat problems. In Sat, we are given a formula in conjunctive normal form (consisting of m clauses and n variables) and the question is whether the variables can be assigned to 0 and 1 so that all clauses are satisfied. In MaxSat, we are also given a formula in conjunctive normal form and an integer k and the question is whether the variables can be assigned to 0 and 1 so that at least k clauses are satisfied.

Which of the following statements about the Sat and MaxSat problems are true?

a) MaxSat is FPT when parameterized by k.

b) MaxSat can be solved in polynomial time if $k \leq \frac{m}{2}$.

c) Sat is NP-hard even when its clauses contain only 1 variable each.

d) Sat is FPT when parameterized by the number of variables.

Clearly indicate which statements are true and which are not. (You get 3 points if you correctly determine the validity of each statement; you get 1 point if there is exactly one mistake and 0 points otherwise.)

Question 3: Randomized Algorithms **(5 Points)**

Consider the following randomized algorithm for generating a 1 with probability 1/3 using independent flips of a fair coin.

```
Algorithm OneWithProb1/3:
if coin-flip = 0
    return 0
else
    return 1 – OneWithProb1/3()
```

a) (3 points)

Argue using a series formula that OneWithProb1/3 returns 1 with probability 1/3.

b) (2 points)

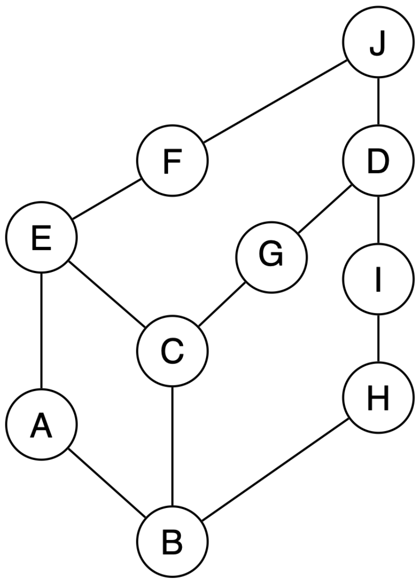
What is the exact expected number of coin flips in one call of OneWithProb1/3?

Question 4: Structural Decompositions and Algorithms **(10 Points)**

a) (5 Points) Recall that an *independent set* of a graph $G = (V, E)$ is a subset $S \subseteq V$ of vertices that are pairwise non-adjacent, that is, $vw \notin E$ for any two vertices $v, w \in S$. The Maximum Independent Set problem asks for the cardinality of a maximum-size independent set in a given graph G.

   Prove that Maximum Independent Set is fixed-parameter tractable when parameterized by the treewidth of the input graph.
   Hint: You may use tractability results for monadic second-order logic model checking presented in the lecture.

b) (5 Points) Determine the treewidth of the graph shown below by proving matching upper and lower bounds.
   Hint: Constructing a tree decomposition is not the only option for proving an upper bound.



Question 5: Linear Programming (5 Points)

a) (5 Points)

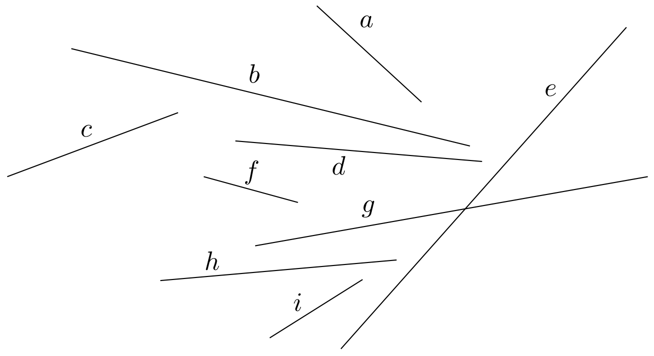
Given an unweighted, undirected simple graph $G = (V, E)$ with $n = |V|$ nodes. We want to delete exactly $k \leq n$ nodes together with their incident edges, such that the number of deleted edges is minimized.

Formulate this problem as a MILP. Describe the used variables together with their domains, formulate the objective function, and state the required constraints. To ease modeling you may assume that $V = \{1, \dots, n\}$.

Question 6: Geometric Algorithms (10 Points)

a) (4 Points)

Consider the following instance for the line segment intersection problem from the lecture. Assume the sweep line ℓ moves from left to right and denote the left endpoint of any segment s as s_0 and its right endpoint as s_1.



Answer the following four questions:

(i) What is the ordered sweep line status *after* the event i_1 has been processed?

(ii) Which intersection tests are performed at event d_0?

(iii) Which intersection tests are performed at event f_1?

(iv) At which event is the intersection between e and g detected?

b) (6 Points)

Let $P \subset \mathbb{R}^2$ be a set of $n \geq 3$ points in general position (no three points are collinear and no four points lie on a common circle). Let $p \in P$ be a point and let δ be the Euclidean distance of a nearest neighbor of p, i.e., $\delta = \min\{||p - r|| \mid r \in P \setminus \{p\}\}$.

Prove or disprove the following claim.

> Let D be the disk centered at p with radius $\delta/2$. Then the interior of D lies completely inside the Voronoi cell $\mathcal{V}(p)$ and the boundary of D touches the edges of $\mathcal{V}(p)$ defined by p and the nearest neighbors of p.