TU Wien
Institute of Logic and Computation
Algorithms and Complexity Group

186.814 Algorithmics VU 6.0
Exam (winter term 2018/19)
January 14, 2019

Insert the following information in clearly readable block letters:

First name: ______ Last name: ______

Student ID: ______ Signature: ______

Please put your student ID on the desk in front of you.

You may write the solutions only to the specification sheets provided by the supervision. You are not allowed to use your own paper. Please use indelible pens (no pencils!).

It is forbidden to use calculators, cellphones, tablets, digital cameras, scripts, books, notes, solved exercises or similar resources.

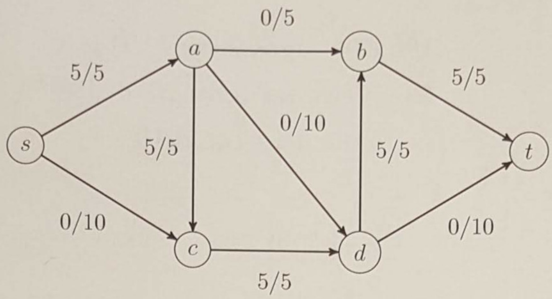
Question	Points	(max)	Question	Points	(max)
1		(10)	4		(10)
2		(10)	5		(10)
3		(5)	6		(5)

Total: ______ (50)

Good luck!

Question 1: Flows and Matchings **(10 Points)**

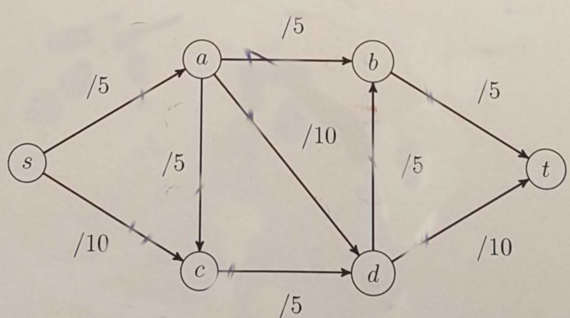
a) Consider following flow network $N = (V, E, s, t, c)$ with a flow f, where each edge e is labeled by $f(e)/c(e)$.



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

b) Indicate in the residual graph an augmenting path P. (1 point)

c) Enter below a maximum flow obtained via the augmenting path P. (1 point)



d) Indicate in the drawing above all minimum cuts. (2 points)

Question 2: Mixed Integer Linear Programming **(10 Points)**

Your task is to find a parking plan for a set of cars $K = \{1, \dots, k\}$ with lengths $\lambda_i \in \mathbb{R}_{>0}$, $i \in K$. Parking is organized in lines $P = \{1, \dots, p\}$. The length of a parking line is specified by the sum of the lengths of the parking cars and may not exceed a specified constant $L \in \mathbb{R}_{>0}$. However, to allow for more flexibility this restriction may be violated in up to θ parking lines. A solution consists of an assignment of cars to parking lines. The goal is to balance the lengths of the parking lines as well as possible. More specifically, we want to minimize the difference of the length of a longest and a shortest line.

Provide a (mixed) integer **linear** programming model for this problem. Describe the used **variables**, formulate the **objective function**, and state the required **constraints** and variable **domains**.

Variables

Objective Function

Constraints

Question 3: Fixed Parameter Tractability (5 Points)

Consider the following problem. Recall that deleting a vertex v from a graph also deletes all edges incident to v.

Instance: An integer k and a graph G where each vertex is either *red* or *blue*.
Parameter: k.
Question: Does there exist a set X of at most k vertices such that deleting X from G results in each blue vertex having at most 4 red neighbors?

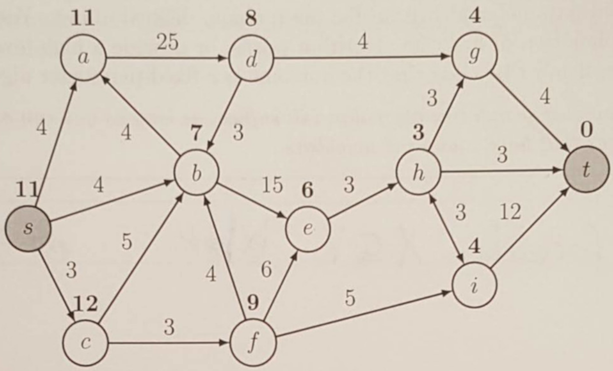
Give a fixed-parameter algorithm for the problem defined above. You may either give a brief explanation of how the algorithm works, or provide a high-level pseudocode; in either case, it must be clear that the outcome is a fixed-parameter algorithm.

Hint: A bounded search tree algorithm will suffice, as long as you can somehow deal with blue vertices that have many red neighbors.

Question 4: A* Search, Randomized Algorithms (10 Points)

a) (4 Points)

Perform the A* algorithm on the following directed graph to find a shortest path from s to t. The bold number above each node x is the heuristic estimate $h(x)$ to reach t and corresponds to the Manhattan distance in the drawing. Edge labels denote the lengths of the arcs, and they are never smaller than the corresponding Manhattan distances of the end points. In case of ties consider nodes in alphabetical order.



List the nodes in the order in which they are first reached by the algorithm. Furthermore, list the nodes that are expanded at each step in the respective order. Last but not least, what is the shortest path and its length?

b) (2 Points)

Can you be sure that A* search yielded here indeed a shortest path? Argue why or why not without considering each node individually.

c) (2 Points)

Does the above graph fulfill monotonicity? Argue why or why not without considering each arc individually.

d) (2 Points)

Which of the following statements are correct?

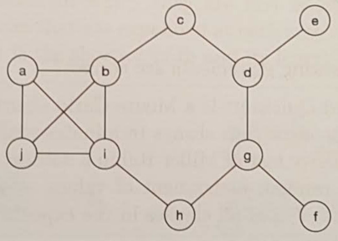
- ☐ Randomized Quicksort is a Monte Carlo algorithm.
- ☐ A Las Vegas algorithm always terminates with an optimal/correct solution.
- ☐ In the primality test of Miller-Rabin a witness proves a number to be prime.
- ☐ A uniform random assignment of values to variables of a 3-SAT instance fulfills at least 7/8 of all clauses in the expected case.

Check exactly the boxes that correspond to a true statement. (You get 2 points if you check exactly the right boxes and no wrong box; if you are wrong in one case, you get 1 point, otherwise 0 points.)

Question 5: Treewidth **(10 Points)**

a) (5 points) The MAXIMUM CLIQUE problem asks for a clique of maximum size in a given graph G. Show that MAXIMUM CLIQUE is fixed-parameter tractable parameterized by the treewidth of the input graph. *Hint: Use monadic second-order logic!*

b) (5 points) Determine the treewidth of the graph depicted below (and prove that your answer is correct).

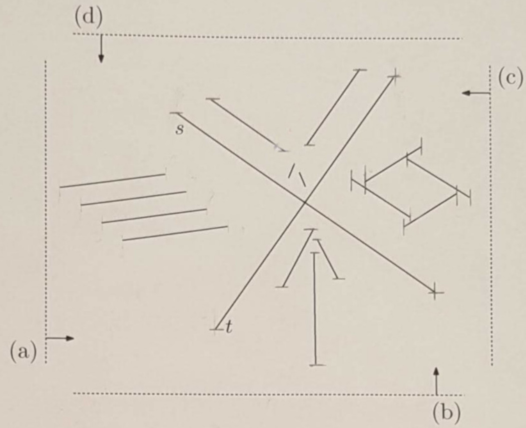


Question 6: Geometric Algorithms

(5 Points)

a) (2 points)

Consider the sweep-line algorithm for detecting line segment intersections from the lecture and the following set of line segments. You can choose one of the four indicated sweep directions.



- For which direction does the algorithm recognize the intersection point between segments s and t the earliest (in terms of the number of processed sweep events)?

  - ☐ (a) left to right
  - ☐ (b) bottom to top
  - ☐ (c) right to left
  - ☐ (d) top to bottom

- Mark by a circle the corresponding event point at which the intersection between s and t is detected according to your selected sweep direction.

b) (3 points)

Consider the following statements:

- ☐ For every set P of $n \geq 2$ points the Voronoi cells of the two closest points are neighbors.
- ☐ The Delaunay triangulation for $n \geq 3$ points (no four on a common circle) has at most $2n - 5$ faces.
- ☐ For every vertex v of degree 1 in a planar graph stored as a doubly-connected edge list it is true that `face(edge(`v`))` = `face(twin(next(edge(`v`))))`.
- ☐ For a point set P of n points with smallest enclosing disk D and a point q drawn uniformly at random from P, the probability that D is the smallest enclosing disk of $P \setminus \{q\}$ is at least $1 - \frac{2}{n}$.

Check exactly the boxes that correspond to a true statement. (You get 3 points if you check exactly the right boxes and no wrong box; you get 1 point if there is exactly one mistake and 0 points otherwise.)