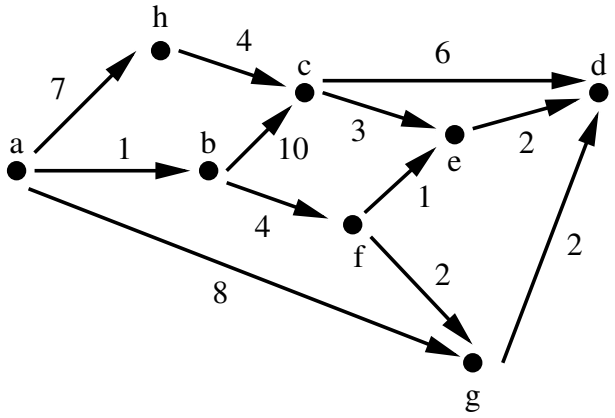
universität innsbruck

Discrete Structures/Mathematics | WS 2019/2020 | LVA 703070/703015

EXAM Test. (Each exercise corresponds to 1 cross in the PS) | January 22nd, 2020

This exam consists of three regular exercises (1–3). The time available is 1 hour and 45 minutes (105 minutes). The available points for each item are written in the margin. There are 60 points in total for the regular exercises. In addition, there is a bonus exercise (4∗) worth 20 points. You need at least 30 points to pass.

1 Let the weighted directed graph $G$ with set of vertices $V = \{a, b, c, d, e, f, g, h\}$ be:




(a) Give a matrix representation $M$ of $G$.

[5] Taking nodes in alphabetic order of their names, we obtain (the rather sparse) matrix $A$ where $A_{ij}$ is the weight of the edge from vertex $v_i$ to $v_j$:

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 8 & 7 \\ 0 & 0 & 10 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 6 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

(b) Compute a shortest path $\pi$ from node $a$ to node $d$ using the shortest path algorithm

[5] for dags, giving at least two intermediate stages of the algorithm.

We give the stages in order, listing the nodes with their predecessor and distance from a in the shortest path until then, and underlining the next selected node (in topological order):

- $a : \underline{0}$;
- $a \to b : \underline{1}$, $a \to h : 7$, $a \to g : 8$;
- $a \to h : \underline{7}$, $a \to g : 8$, $b \to c : 11$, $b \to f : 5$;
- $a \to g : 8$, $b \to c : \underline{11}$, $b \to f : 5$;
- $a \to g : 8$, $b \to f : \underline{5}$, $c \to e : 14$, $c \to d : 17$;

- $f \to g : 7$, $f \to e : \underline{6}$, $c \to d : 17$;
- $f \to g : \underline{7}$, $e \to d : 8$;
- $e \to d : \underline{8}$;

from which we read off that $a \to b \to f \to e \to d$ is the shortest path from $a$ to $d$, with weight 8.

(c) [5] Let $R$ be the binary relation $R$ on $V$ underlying graph $G$, and let the function $f : V \to \mathbb{N}$ be defined by $f(v) = \Sigma_{wRv} f(w)$, i.e. the $f$-value of node $v$ is the sum of the $f$-values of all nodes $R$-related to $v$. Give $R$ (as set of pairs), explain why $f$ is a well-defined function, and compute $f(e)$.

$R = \{(a,h),(a,b),(a,g),(h,c),(b,c),(b,f),(c,d),(c,e),(f,e),(f,g),(e,d),(g,d)\}$. Since $R$ does not have infinite descending chains, it is well-founded, allowing to define functions by well-founded recursion on $R$. In this case, the $f$-value of a node only depends on the values $R$-related to the node, so $f$ is well-defined. In particular $f(a) = \Sigma_{wRa} f(w) = 0$, since there is no $w$ such that $w\ R\ a$. It easily follows (formally: using a proof by well-founded induction) from this that in fact $f(w) = 0$ for all $w$. In particular $f(e) = 0$.

Alternatively, without using the observation, one can directly compute $f(e) = f(c) + f(f) = f(h) + f(b) + f(b) = 3 \cdot f(a) = 3 \cdot 0 = 0$.

(d) [5] Let $U$ be the *undirected* graph associated to $G$, obtained by forgetting the direction of the edges of $G$. Compute a minimal spanning tree $T$ of $U$ using Kruskal's algorithm, giving at least two intermediate stages of the algorithm.

We list the stages (partitionings of the set of nodes) of the algorithm, after adjoining (in order of increasing weight) edges:

- $\{\{a\},\{b\},\{c\},\{d\},\{e\},\{f\},\{g\},\{h\}\}$;
- $\{\{a -_1 b\},\{c\},\{d\},\{e -_1 f\},\{g\},\{h\}\}$;
- $\{\{a -_1 b\},\{c\},\{e -_1 f -_2 g -_2 d\},\{h\}\}$;
- $\{\{a -_1 b\},\{c -_3 e -_1 f -_2 g -_2 d\},\{h\}\}$;
- $\{\{a -_1 b -_4 f \text{ and } h -_4 c -_3 e -_1 f -_2 g -_2 d\}\}$;

Note the solution is not unique. E.g. the edge between $g$ and $d$ can be exchanged for the edge between $e$ and $d$ (of the same weight, so not changing the overall weight).

[5] 2 (a) Let $A = \{a,b,c,d,e\}$ and $B = \{0,1\}$. Compute the number $s$ of subsets of $A$, the number $t$ of subsets of size 2 of $A$, and the number $f$ of (total) functions from $A$ to $B$. (Note that the number of *partial* functions from $A$ to $B$, i.e. functions where $f$ is undefined on some inputs, is $3^5$, as we then have one extra option for every input.)

$s = 2^5 = 32$, $t = \binom{5}{2} = \frac{5!}{3!\cdot 2!} = 10$, $f = 2^5 = 32$.

(b) [5] Let $p = 17$, $q = 12$, $a = 3$ and $b = 4$. Compute the inverse $p'$ of $p$ modulo $q$, and an $x$ such that $x \equiv a \pmod{p}$ and $x \equiv b \pmod{q}$. Indicate which algorithm(s) you use and give at least two intermediate stages for both computations.

Since the gcd of 17 and 12 is 1 (17 is a prime number), we may compute $p'$ using Bézout's lemma and the gcd algorithm:

$$\begin{aligned} 17 &= 1 \cdot 17 + 0 \cdot 12 \\ 12 &= 0 \cdot 17 + 1 \cdot 12 \\ 5 &= 1 \cdot 17 - 1 \cdot 12 \\ 2 &= -2 \cdot 17 + 3 \cdot 12 \\ 1 &= 5 \cdot 17 - 7 \cdot 12 \end{aligned}$$

computing modulo 12 the right-hand side simplifies to $5 \cdot 17$, so we see that $1 \equiv 5 \cdot 17 \pmod{12}$, hence $p' = 5$ is inverse to 17 modulo 12. Using this inverse we may compute, using the RSA-version of the Chinese Remainder Theorem, directly that $x = 3 + 17 \cdot (5 \cdot (4-3) \mod 12) = 88$. One easily verifies that $17 \cdot 5 \equiv 1 \pmod{12}$, and $88 \equiv 3 \pmod{17}$, and $88 \equiv 4 \pmod{12}$, as desired.

(c) Suppose the complexity $T$ of some algorithm $A$, as a function of the size $n$ of its input, [5] is given by $T(n) = T\left(\frac{n}{2}\right) + 4 \cdot n$ if $n > 4$, and 5 otherwise. Determine a closed-form expression $e$ such that $T(n) \in \Theta(e)$, and explain what the latter notation means.

The Master theorem applies with $a = 1$, $b = 2$ and $s = 1$. By the third case of the theorem, we have $e = n$ and $T(n) \in \Theta(n)$. The notation means that the asymptotic complexity $T$ is linear, i.e. $T$ is asymptotically bounded both from below and above by a function linear in $n$.

[5] (d) Show that the set $\{M\#x \mid \text{TM } M \text{ rejects input } x\}$ is not recursive, where you may assume that (the code of) $M$ and $x$ are bit-strings.

Let $A = \{M\#x \mid \text{TM } M \text{ rejects input } x\}$. By a reduction from the membership problem $MP$, i.e. we show $MP \leq A$. We let the reduction $f$ map $M\#x$ to $M'\#x$ where the TM $M'$ is obtained from $M$ by swapping its reject and accept states. Then $M\#x \in MP$ iff $M$ accepts $x$ iff $M'$ rejects $x$ iff $M\#x \in \{M\#x \mid \text{TM } M \text{ rejects input } x\}$. (More explicitly, distinguishing cases on whether the string is or is not in $MP$: if $M\#x \in MP$ then $M$ accepts $x$, hence $M'$ rejects $x$ by definition of $M'$, so $M'\#x \in A$, i.e. $f(M\#x) \in A$. if $M\#x \notin MP$ then $M$ rejects or loops on $x$, hence $M'$ accepts resp. loops on $x$ by definition of $M'$, so $M'\#x \notin A$, i.e. $f(M\#x) \notin A$.)

Alternatively, we can adapt the proof by 'diagonalising away' as given for $HP$ and $MP$ to show that $A$ is not recursive. That is, first one defines a behaviour of a TM to be either rejecting or not rejecting. so that the complement *cd* of the behaviour of the diagonal is not the behaviour of any TM, and next one shows that if there were some total TM $K$ such that $L(K) = A$, then we *could* build a TM $CD$ (using $K$ as a component) exhibiting behaviour *cd*. More precisely, we may construct $CD$ such that on input $x$ it feeds $M_x\#x$ into $K$ and if that accepts (i.e. if $M_x$ rejects $x$) then $CD$ accepts, and otherwise (i.e. if $M_x$ accepts or loops on $x$) then $CD$ rejects. Thus, if $M_x$ rejects $x$ then $CD$ does not reject $x$ (it accepts it), and if $M_x$ does not reject $x$ (it accepts or loops on it) then $CD$ rejects it, i.e. $CD$ would exhibit behaviour $CD$.

[20] 3 Determine whether the following statements are true or false. Every correct answer is worth 2 points. For every wrong answer 1 point is subtracted, provided the total number of points is non-negative.

statement

---

$(A^\ell)_{ii}$ is the number of cycles of length $\ell$ on node $v_i$, for $A$ the adjacency matrix of a directed graph with nodes $v_1, \ldots, v_n$.

Well-founded orders are closed under union.

For a partial order, every minimal element is least.

The specification $f(0) = 1$ and $f(n) = f(f(n-1)) - 1$ if $n > 0$, defines a function $f : \mathbb{N} \to \mathbb{N}$, with $-$ the usual subtraction of integers.

By inclusion/exclusion, $\#(\bigcup_{i=1}^{k} A_i) \leq \sum_{I \subseteq \{1,\ldots,k\},\ \#(I) \text{ odd}} \#(\bigcap_{i \in I} A_i)$, for sets $(A_i)_{1 \leq i \leq k}$.

If $c \cdot a \equiv c \cdot b \bmod n$, then $a \equiv b \bmod n$, for all natural numbers $a, b, c, n \geq 2$.

For all functions $f, g$ on $\mathbb{N}$, $f \in O(g)$ or $g \in O(f)$ or both.

If there is an injective function from $A$ to the set $\{n \in \mathbb{N} \mid n \geq 10\}$, then $A$ is finite.

If $L = L(M)$ for some TM $M$, then $L$ or $\sim L$ is recursive.

For all DFAs $A$ there exist strings $x, y$ and a state $q$ such that $\hat{\delta}(q, x) = \hat{\delta}(q, y)$ and $x \neq y$.

YES, NO, NO, NO, YES, NO, NO, NO, NO, YES.

4* [10] (a) Let $f : \mathbb{N} \to \mathbb{N}$ be defined by $f(0) = 0$ and and $f(n) = f(n-1) + 3 \cdot n \cdot (n-1) + 1$ if $n \geq 1$. Show that $\forall n \in \mathbb{N}$, $f(n) = n^3$.

Since $n - 1 < n$ and $<$ is well-founded, it suffices to verify that the equalities hold, when substituting $n \mapsto n^3$ for $f$. This is trivial for the first and easy for the second: $n^3 = n^3 - 3 \cdot n^2 + 3 \cdot n - 1 + 3 \cdot n \cdot (n-1) + 1 = (n-1)^3 + 3 \cdot n \cdot (n-1) + 1$.

Alternatively, it can be proven by induction on $n$, or also by repeated expansion and seeing a pattern.

[10] (b) Prove that there is a bijection between the set $\mathbb{N}$ of natural numbers and the set $P$ of palindromes over $\{0, 1\}$, i.e. $P$ contains all bit-strings $x$ that are the same as their reverse.

The function mapping $n$ to the palindrome $0^n$ is an injection, and so is the function mapping a palindrome $x$ to $(1x)_2$, the value of the bit-string $1x$. We conclude by the theorem of Schröder–Bernstein.

(Note that there are many possibilities for the injections. For instance, $n \to 1^n$ for the former and $x \to (x[0{:=}2])_3$ for the latter, where $x[0{:=}2]$ denotes the string obtained from $x$ by replacing all 0s by 2s and $(y)_3$ denotes the value of the string $y$ in ternary.)