MATEMATISKA INSTITUTIONEN
STOCKHOLMS UNIVERSITET
Avd. Beräkningsmatematik
Examinator: Lars Arvestad
Lecturer and exam designer: Marc Hellmuth

Tentamensskrivning i
Algoritmer och komplexitet
7.5 hp
2021-03-19

General Information (that was already listed at the homepage)

It is not allowed to share answers!

- All single pages of your solutions must be clearly marked with your StudentID and name (no anonymity code).
- You have 2 hours to solve the exam and you get additional 20 minutes to upload your solutions.

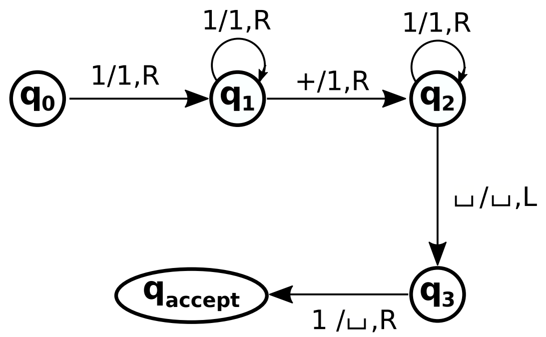
  Hence, upload your solutions **not later than April 26 - 3:20pm** at the bottom of the Kurser "DA3004" homepage under PRE-EXAM or using the link https://kurser.math.su.se/mod/assign/view.php?id=62321.

  ONLY IN CASE something does not work with the upload, you can also send your solutions via email to `marc.hellmuth@math.su.se`.
- In case you have questions during the exam, use the zoom-link https://stockholmuniversity.zoom.us/j/68673538207
- In total, you can get 100 points and you need to get at least 55 points to pass the exam.

Problem	1	2	3	4	5	6	7	8	TOTAL
Points									

Problem 1 (*Turing Machine*) 5+5=10p

Given is the following Turing machine M with initial state q_0 and blank symbol "⌣" (in simplified finite state representation):



(a) Which language does this Turing maching recognize?

(b) What is final string on the tape for accepted strings?

1a) the language that contains all strings of the form: $\underbrace{1..1}_{m \text{ times}} + \underbrace{1...1}_{n \text{ times}}$, integ. m, n with $m \geq 1$, $n \geq 0$

b) "+" replaces by "1" & last "1" removed (if existent)

so we get concatenation of all "1"

$$\underbrace{1..1}_{m \text{ times}} + \underbrace{1...1}_{n \text{ times}} \xrightarrow{TM} \underbrace{1..1}_{m \text{ times}} \underbrace{1...1}_{n \text{ times}}$$

Problem 2 (*Runtime*) 5+10=15p

(a) Show that $T(n) = n - 1 \in \Theta(n)$ (in "big-Theta").

(b) Consider the following algorithm for searching an integer x in a sorted list A of size n.

```
FIND(integer x, list A of size n)
L = 0
R = n − 1
while L ≤ R do
    m = ⌊(L+R)/2⌋
    if A[m] < x then
        L = m + 1
    else if A[m] > x then
        R = m − 1
    else return m
return "x not found"
```

Here, $\lfloor \frac{L+R}{2} \rfloor$ is the floor function that returns the greatest integer less than or equal to $\frac{L+R}{2}$. Determine the runtime in big-O notation (best possible bound) assuming that all basic operations (e.g. return, addition, mulitplication, cases, floor, assignments, ...) can be done in constant time. Explain your results.

1a) to show $T(n) \in \Omega(n)$
$\in O(n)$

$T(n) = n-1 \leq n \quad \forall n \geq 1 \Rightarrow n-1 \in O(n)$

$cn \leq n-1 \Leftrightarrow c \leq \frac{n-1}{n} = 1 - \frac{1}{n}$

$\Rightarrow$ choose $c = \frac{1}{2}$

$\Rightarrow \frac{1}{2} n \leq n-1 \quad \forall n \geq 2 \Rightarrow (n-1) \in \Omega(n)$

$\Rightarrow n-1 \in \Theta(n)$

b) this is just a sketch of solution:

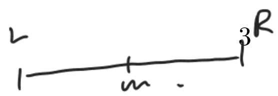
all steps constant $\rightarrow$ Q: how long does WHILE run?

in each step of while $m \simeq \frac{L+R}{2}$ & start with $L+R=n$

$\Rightarrow T(n) = T(1) + T(\frac{n}{2})$

$\vdots$

$= N(T(1)) + T(\frac{n}{2^N})$ $\rightarrow$ while loop stops after N steps.



in each step halved

$\Rightarrow N = \log_2(n) \Rightarrow \log_2(n) T(1) + \underbrace{T(\frac{n}{2^{\log_2(n)}})}_{= T(1)}$

$\rightarrow O(\log_2(n))$ □

Problem 3 (*Complexity*) 7.5+7.5=15p

Suppose we are given a decision problem A that has as one of the inputs an arbitrary undirected graph. Let $\mathcal{T}$ denote the set of all graphs that are trees and $\mathcal{F}$ denote the set of all graphs that are forests.

(a) Assume we have shown NP-hardness of A by reduction from 3-SAT by constructing a special disconnected forest $G \in \mathcal{F}$. Explain shortly, if this implies that A is NP-hard for the class of graphs in $\mathcal{T}$?

(b) Assume we have shown NP-hardness of A by reduction from 3-SAT by constructing a special tree $G \in \mathcal{T}$. Explain shortly, if this implies that A is NP-hard for the class of graphs in $\mathcal{F}$?

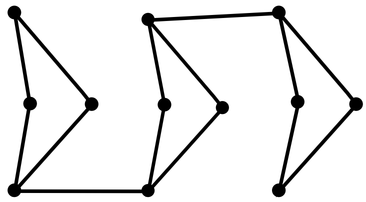
a) We don't know whether the problem remains NP-hard for graphs in $\mathcal{T}$, since NP-hardness was shown only for graphs that are <u>not</u> contained in $\mathcal{T}$.

b) Yes, this implies NP hardness for graphs in $\mathcal{F}$ since we have used a reduction to some $G \in \mathcal{T} \subseteq \mathcal{F}$

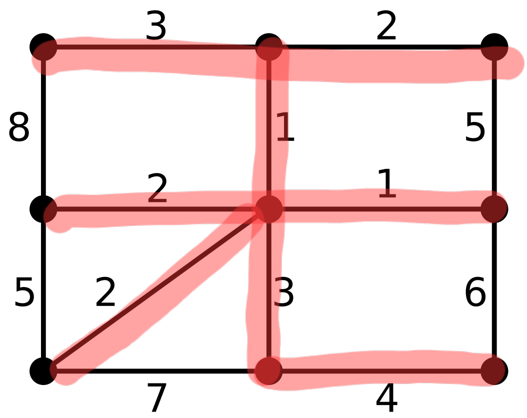
Note: $\mathcal{T}$ = all trees
$\mathcal{F}$ = all forests
& every tree is a (connected) forest
$\Rightarrow \mathcal{T} \subseteq \mathcal{F}$.

Problem 4 (*Trees*) 5+5=10p

(a) How many spanning trees has the following graph? Shortly explain your results.



(b) Draw into the following graph a minimum spanning tree that you find by applying Kruskal's algorithm.



a) each  has 4 spanning trees

each edge connecting 2  must be contained in spanning tree

$\Rightarrow 4 \cdot 4 \cdot 4 = 64$ sp. trees.

Problem 5 (*Approximation Algorithms*) 10p

In the lecture, the following 2-approximation algorithm for the Vertex Cover Problem in general graphs was provided:

GREEDY_VC2($G = (E, V)$)
$C = \emptyset$, $E' = E$
while $E' \neq \emptyset$ **do**
  $e = \{u, w\}$ some edge in E'
  $C = C \cup \{u, w\}$
  Remove all edges incident to u and w from E'
return C

Show that, in general, there is no constant ρ with $1 \leq \rho < 2$ such that algorithm GREEDY_VC2 is a ρ-approximation algorithm.

give contrexample: e.g. $K_{n,n}$

has VC C

Greedy_VC2 may return edges:

which gives VC of size $2 \cdot |C|$

$\Rightarrow$ no ρ, $1 \leq \rho < 2$ exists!

Problem 6 (*Greedy and Matroids*) 5+10=15p

A *matching* in an undirected graph G is a subset $M \subseteq E(G)$ of edges such that no two edges in M share a common vertex, i.e., $e \cap f = \emptyset$ for all distinct $e, f \in M$.

Our goal is to find a matching M of maximum size in a given undirected graph G.

(a) Define the independence system $(E, \mathbb{F})$ that describes this problem and also prove that $(E, \mathbb{F})$ is an independence system.

(b) Prove that a greedy algorithm will in general not optimally solve this problem by showing that $(E, \mathbb{F})$ is not a matroid.

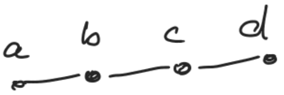
$E := E(G)$

$\mathbb{F} := \{ E' \subseteq E : E' \text{ is matching} \}$.

M1) $\emptyset \in \mathbb{F}$: yes since no edges in $\emptyset$ share common vertex $\Rightarrow$ $\emptyset$ is matching.

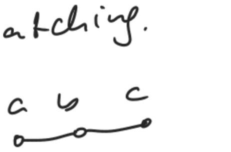
$E \in \mathbb{F}$, $E' \subseteq E$ $\Rightarrow$ since no two edges in E share common vertex $\Rightarrow$ no two edges in E' share common vertex since $E' \subseteq E$ $\Rightarrow$ E' is matching $\Rightarrow$ $E' \in \mathbb{F}$.

M2) Exmpl: G



$\Rightarrow$ $E = \{\{a\,b\}, \{c\,d\}\}$ matching
$E' = \{\{b\,c\}\}$ matching.

$|E'| < |E|$ but



no matching

$\Rightarrow$ exchange property not satisfied

$\Rightarrow$ $(E, \mathbb{F})$ no matroid. $\square$

Problem 7 (*Dynamic Programming*) 5+10=15p

You are given an exam with questions numbered $1, 2, 3, \ldots, n$. Each question i is worth p_i points. *You must answer the questions in order, but you may choose to skip some questions.* The reason you might choose to do this is that even though you can solve any individual question i and obtain the *entire* p_i points, some questions are so frustrating that after solving them you will be unable to solve *any of the following* f_i questions.

By way of example, in the tabular below the values p_i and f_i are given for questions $1-5$. If you decide to solve the 2nd question with $f_2 = 2$ and $p_2 = 2.5$, you will get all 2.5 points, but you are not able to solve questions 3 and 4.

(a) Assume that the values p_i and f_i are as given in the following tabular.

question	1	2	3	4	5
p_i	1	2.5	3	1	3
f_i	0	2	1	1	0

Which questions should you choose? *(no proof of correct solution needed here)*

(b) Suppose that you are given the p_i and f_i values for all n questions as input. Give a dynamic programming solution to compute the maximum number of points one can achieve.

a) choose question 1, 3, 5

b) $S(i)$ = opt value for questions i through n

$\Rightarrow$ Question i is either included in the opt choice or not.

IF i included $\Rightarrow$ $S(i) = p_i + S(i + f_i + 1)$
i not included $\Rightarrow$ $S(i) = S(i+1)$

init $S(0), S(1) \ldots S(n+1) \ldots S(n + f_n + 1) = 0$

FOR ($i = n..1$)
$S(i) = \max\{S(i+1), p_i + S(i + f_i + 1)\}$

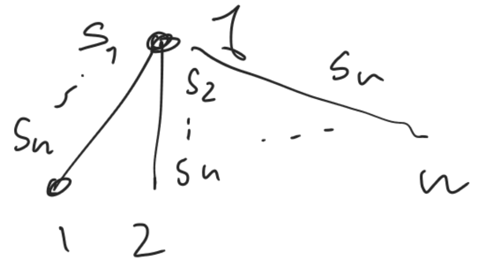
Problem 8 (*Suffix Trees*) 5+5=10p

(a) For an arbitrary integer $n \geq 1$, provide an alphabet Σ and construct a string $S = s_1 \dots s_n \in \Sigma^n$ such that the root of the resulting suffix tree has as children only leaves. Explain your result.

(b) Draw the suffix tree for the string $S = BACBAAA\$$.

$S = s_1 \dots s_n$ st $s_i \neq s_j \ \forall i \neq j$

a)



[Explain your results]

b)