MATEMATISKA INSTITUTIONEN
STOCKHOLMS UNIVERSITET
Avd. Beräkningsmatematik
Examinator: Marc Hellmuth

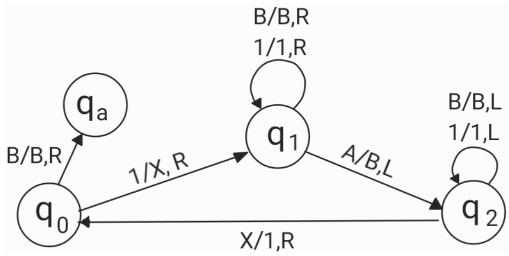
Tentamensskrivning i
Algoritmer och komplexitet
7.5 hp
2021-10-21

General Information

- Time 08:00 - 13:00
- No books, internet, smartphones, notebooks etc allowed.
- In total, you can get 100 points and you need to get at least 60 points to pass the exam.

Problem 1 (*Turing Machine (TM)*) 4+4=8p

Given is the following Turing machine M with initial state q_0 and acceptance state q_a (in simplified finite state representation):



Let $\Sigma = \{1, A\}$.

(a) Explain which language $L \subseteq \Sigma^*$ is recognized by M?

Hint: Try the strings: $s = 1A$; $s = 11AAA$; $s = 11A$

(b) In general, what is the final string provided by M when the input is a string $s \in L \subseteq \Sigma^*$ that is accepted by M?

Problem 2 (*Runtime*) (3+3+3)+(2+2+2)=15p

(a) Show *in detail* for

(i) $f(n) = \log_2(n)$ and $g(n) = \log_{16}(n)$

(ii) $f(n) = n^2$ and $g(n) = 2^n$.

(iii) $f(n) = 6^n$ and $g(n) = 3^n$

which of the relationships $f(n) \in O(g(n))$, $f(n) \in \Omega(g(n))$ and $f(n) \in \Theta(g(n))$ are satisfied and which are not.

The term "*in detail*" means, that *for all* assumptions you make also proofs should be provided, e.g., you must prove that $n^2 \geq 2^n$ (the statement "*this is easy to see*" will not be accepted). However, you can apply rules for logarithm as well as power laws without proofs.

(b) Consider the following recursive algorithm.

S(integer n)

if $n = 0$ **then return** 1

else return $S(n-2) + 2 \cdot n$.

(i) Apply this algorithm with input $n = 8$ and specify the computed values $S(i)$ in each step of the recursion until the algorithm terminates. What is the final computed value?

(ii) For which input values does the algorithm terminate? Show that the returned final value will always be an odd integer; provided the algorithm terminates.

(iii) Determine the runtime in big-O notation (best possible bound) assuming that basic operations as return, addition and multiplication can be done in constant time. Explain your results.

Problem 3 (*Complexity*) 3+7=10p

(a) Define the class NP. Explain briefly what can be said about the elements in NP $\setminus$ P.

(b) Let $\mathcal{C}$ be a set of subsets of a finite set S and $k \geq 1$ be a positive integer. A *hitting set* for $\mathcal{C}$ is then a subset $S' \subseteq S$ such that $S' \cap C \neq \emptyset$ for all $C \in \mathcal{C}$, i.e., S' contains at least one element from every element $C \in \mathcal{C}$.

Consider the following problem HittingSet:

> Given a set $\mathcal{C}$ of subsets of a finite set S and a positive integer $k \geq 1$.
> Does S contain a *hitting set* S' for $\mathcal{C}$ with $|S'| \leq k$?

Prove that this problem is NP-complete by reduction from the Vertex-Cover-Problem (VCP).

To recap the NP-complete VCP: *Determine if there is a subset $W \subseteq V$ of size $|W| \leq \ell$ such that for all edges $e = \{u, v\} \in E$ at least one of u and v is contained in the set W for a given undirected graph $G = (V, E)$ and non-negative integer ℓ.*

Problem 4 (*Graphs and Trees*) (5+2.5)+7.5=15p

(a) The following statement can be assumed to be true: *If $G = (V, E)$ is a tree, then $|E| = |V| - 1$.*

(i) Let G be a connected undirected graph.
Prove the converse: *If $|E| = |V| - 1$, then $G = (V, E)$ is a tree*

(ii) Is the statement in 4a.i still valid for disconnected graphs?

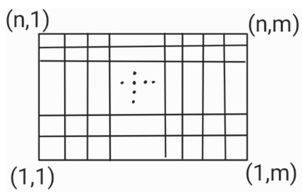
(b) Recap Kruskal's algorithm.

Kruskal($G = (V, E), w \colon E \to \mathbb{R}$) // $m = |E|$
sort edges such that $w(e_1) \leq w(e_2) \cdots \leq w(e_m)$
$F = \emptyset$, $T = (V, F)$
for $i = 1, \ldots, m$ **do**
    if $(V, E \cup \{e_i\})$ is acyclic **then**
        $T = (V, E \cup \{e_i\})$
return T

Explain, in words, how to use Kruskal's algorithm to compute the number of connected components in an undirected graph.

Problem 5 (*Dynamic Programming and Shortest Paths*) 7.5+2.5=10p

Shown is an $n \times m$ grid where the vertices are labeled with corresponding coordinates (i, j), $1 \leq i \leq n$, $1 \leq j \leq m$.



(a) Give a recursive formula for computing the number of simple shortest paths from $(1, 1)$ to (n, m). Explain shortly why your formula works.

(b) What is the number of simple shortest paths from the origin $(1, 1)$ to the vertex with coordinates $(3, 4)$?

Problem 6 (*Greedy and Matroids*) 4+(5+5)=14p

In the following, E is a finite set and $\mathbb{F}$ a collection of subsets of E.

(a) Show that all basis elements of a matroid must have the same size.

(b) Consider the following problem. Let S be a collection of students and let $c\colon S \to \{1, \ldots, m\}$, $m \geq 1$ be a map that assigns to every student $s \in S$ a particular skill $c(s)$.
The simple aim is to find a maximum-sized subset $S' \subseteq S$ of students such that all students in S' have different skills, i.e., $c(s) \neq c(s')$ for all distinct $s, s' \in S'$

(i) Design a polynomial-time greedy algorithm to find such a subset $S' \subseteq S$ of maximum size.

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

Problem 7 (*Approximation Algorithms*) 1+(2+7)=10p

(a) Define when an algorithm is a ρ-approximation algorithm for an optimization problem, $\rho \in \mathbb{R}_{\geq 1}$

(b) A matching M of an undirected graph $G = (V, E)$ is a subset $M \subseteq E$ such for all two distinct edges $e, f \in M$ it holds that $e \cap f = \emptyset$. Given is the following algorithm to compute an inclusion-maximal matching M of G.

```
GREEDY_MATCHING(G = (V, E))
M = ∅
while E ≠ ∅ do
    e = {u, w} some edge in E
    M = M ∪ {e}
    Remove all edges incident to u and w from E
return M
```

(i) Show that GREEDY_MATCHING computes an inclusion-maximal matching M of G.

(ii) Show that GREEDY_MATCHING is a 2-approximation algorithm for the problem of computing a matching of maximum size.
Hint: You could consider, for all edges $e = \{u, w\} \in M$, the set $M_e \subseteq M^*$ that contains all edges of a given fixed maximum-sized matching M^* that are incident with u or w and show that $M^* = \cup_{e \in M} M_e$.

Problem 8 (*Balanced Binary Search (AVL) Trees*) 5+4 =9p

(a) What is the best possible asymptotic bound for searching an element in an AVL tree with $n \cdot 2^n$ vertices in a worst case?

(b) Insert the elements 1 , 2 , 3 , 0.5 , 0.2 , 0.7 (in this order) in an initial empty AVL tree. Draw the AVL tree before and after each step where you need to apply a single rotation.

Problem 9 (*Suffix Tree*) 4+ 5 =9p

(a) Argue if it is possible to design a string $s = s_1 \ldots s_n$ with $s_i \neq s_n = \$$ for all $i \in \{1, \ldots, n-1\}$ such that the root of the corresponding suffix tree has only a single child.

(b) Build the suffix tree $\mathcal{T}$ of the string $s = HEJHEJ\$$.
In addition, provide the compressed suffix tree for s.