PŘÍJMENÍ a JMÉNO:
Login studenta:
DATUM:

Závěrečný test z předmětu "Vyčíslitelnost a složitost"

Doba trvání: **90 minut** Max. zisk: **100 bodů**

Obecné pokyny:
Po obdržení testu ihned do pravého horního rohu napište čitelně své příjmení a jméno, svůj login a datum.
K řešení úloh používejte pomocné papíry. Požadovaný *výsledek* zapište odpovídajícím způsobem do odpovídajícího místa přímo *na zadání testu*. V případě, že se vám (mimořádně) stane, že se řešení nevejde do vyhrazeného místa, pište jej na *zvláštní list* – u příslušné úlohy (např. s číslem [8]) pak napište výrazné ZP (tj. zvláštní papír) a výsledek na zvláštním papíru označte zřetelně číslem úlohy (v našem příkladu tedy číslem 8). (Kromě doplněného zadání a případných označených zvláštních listů nakonec odevzdáte i pomocné papíry, aby se odstranily případné pochybnosti o tom, jak jste k výsledkům dospěli.)
Úlohy řešte v libovolném pořadí, rozvrhněte si dobře čas.
Poznámka: Tento ilustrační test má dát hrubou představu o struktuře a obsahu (skutečného) závěrečného zkušebního testu.
Test obsahuje 10 příkladů. Tomu odpovídá níže načrtnutých 10 okruhů - každý okruh bude prověřen jedním příkladem. Pro konkrétnost je zde u každého okruhu jeden z možných příslušných příkladů uveden. (Jen poznamenávám, že pořadí příkladů ve skutečném testu nemusí odpovídat zde zvolenému pořadí okruhů.)
Speciálně upozorňuji, že některé příklady budou předpokládat i pochopení látky, jež byla určena pro referáty na cvičení.

==========

[1] (**10 bodů**)

Okruh: Výpočetní model RAM, časová a prostorová složitost, uniformní (jednotková) a logaritmická míra.

Příklad:

```
        READ
        STORE   N
        LOAD    =2
while:  STORE   temp
        LOAD    N
        JGTZ    body
        LOAD    temp
        WRITE
        HALT
body:   SUB     =1
        STORE   N
        LOAD    temp
        MUL     temp
        JUMP    while
```

Pro uvedený RAM M uveďte co nejlepší (horní) odhad časové složitosti $T_M(n)$ v uniformní (tj. jednotkové) míře a co nejlepší (horní) odhad prostorové složitosti $S_M(n)$ v logaritmické míře:

(v jednotk. míře) $T_M(n) \leq$

(v logaritm. míře) $S_M(n) \leq$

---

[2] (**8 bodů**)

Okruh: pochopení značení $O, o, \Omega, \omega, \Theta$.

Příklad:

Nechť funkce f, g jsou definovány vztahy

$$f(n) = 158n^2 \log n + 2541n^2 + 145n$$

$$g(n) = max\,\{\,0\,,\,0.0005n^3 - 1578n^2\,\}$$

Uveďte všechny platné vztahy typu $f \in X(g)$, $g \in X(f)$, kde za X lze dosadit libovolný ze symbolů O, o, Ω, ω, Θ.

---

[3] (**10 bodů**)

Okruh: Turingův stroj, jeho různé varianty, jejich vzájemná polynomiální simulace, simulace RAMu pomocí TS a naopak.

Příklad:

Uveďte co nejlepší asymptotický horní odhad časové složitosti jazyka L (tj. časové složitosti problému příslušnosti slova k jazyku L), kde

$L = \{w \in \{a, b\}^* \mid w = w^R\}$ (w^R označuje zrcadlový obraz slova w)

přičemž za referenční model algoritmu bereme

a/ standardní Turingův stroj; v tom případě $L \in \mathcal{T}(f(n))$, kde $f(n) =$

b/ Turingův stroj s dvěma hlavami (přechodová funkce je typu $\delta : (Q - F) \times \Gamma \times \Gamma \longrightarrow Q \times \Gamma \times \{-1, 0, 1\} \times \Gamma \times \{-1, 0, 1\}$; hlavy stojí při startu na první buňce vstupního slova, pak se ale mohou "rozjet");
v tom případě $L \in \mathcal{T}(f(n))$, kde $f(n) =$

---

[4] (**12 bodů**)

Okruh: Obecné metody návrhu algoritmů: prohledávání, metoda 'rozděl a panuj', 'greedy' algoritmy, dynamické programování.

Příklad:

Mějme zadánu bezkontextovou gramatiku G; uvažujme problém příslušnosti k jazyku $L(G)$ a níže načrtnutý algoritmus ALG, který ho řeší. ALG ilustruje jistou obecnou metodu návrhu algoritmů.

Kterou ? (Doplňte:)

ALG pro

vstup: slovo $a_1 a_2 \ldots a_n$

postupně vyplňuje "políčka"

$P_{1,1}, P_{2,1}, P_{3,1}, \ldots, P_{n,1}$

$P_{1,2}, P_{2,2}, P_{3,2}, \ldots, P_{n-1,2}$

$P_{1,3}, P_{2,3}, P_{3,3}, \ldots, P_{n-2,3}$

...

$P_{1,n}$

Do každého políčka přiřadí ALG některé neterminály gramatiky G. Jakou podmínkou jsou charakterizovány (tedy co splňují) neterminály, které ALG přiřadí do políčka $P_{i,j}$?

(Doplňte:) neterminál A je přiřazen do $P_{i,j} \iff$

Uveďte asymptotický (horní) odhad časové složitosti vyplnění jednoho políčka:

a také odhad časové složitosti algoritmu ALG:

---

[5] (**8 bodů**)

Okruh: Pojem 'problém', specifikace konkrétních problémů.

Příklad:

Specifikujte problém obchodního cestujícího (ANO/NE verze)

Název: TSP (*problém obchodního cestujícího*) (ANO/NE verze)
Vstup:

Otázka:

Uveďte nějaký (velmi jednoduchý) vstup

pro nějž je odpověď ANO:	pro nějž je odpověď NE:

---

[6] (**8 bodů**)

Okruh: Třídy složitosti problémů ($\mathcal{T}(f(n))$, $\mathcal{S}(f(n))$), třídy PTIME, NPTIME, jejich robustnost vůči (rozumným) výpočetním modelům.

Příklad:

Uveďte vztahy inkluze mezi třídami složitosti $\mathcal{S}(n^2)$, $\mathcal{T}(n)$, $\mathcal{S}(n)$, $\mathcal{T}(n^2)$, jejichž platnost umíte stručně zdůvodnit (pokud možno, toto zdůvodnění také uveďte):

---

[7] **(12 bodů)**

Okruh: Polynomiální převeditelnost problémů. NP-obtížnost, NP-úplnost.

Příklad:

Ukažte, že problém 3-SAT je polynomiálně převeditelný na problém ILP (integer linear programming, tj. celočíselné lineární programování).

Vaším úkolem je tedy načrtnout algoritmus *Alg* s polynomiální časovou složitostí, jenž

očekává na vstupu (doplňte, co):

a pro takový vstup
vydá výstup (doplňte, co je výstupem):

přičemž výstup musí splňovat následující podmínku (ve vztahu ke vstupu):

Stačí, když budete algoritmus ilustrovat na jednoduchém příkladu (z něhož bude zřejmé, jaký výstup *Alg* k zadanému vstupu vyrobí)

---

[8] **(8 bodů)**

Okruh: Vztahy tříd PTIME, NPTIME, PSPACE, NPSPACE, EXPTIME, EXPSPACE. Příklady problémů různé obtížnosti.

Příklad:

O kterých vztazích inkluze mezi třídami PTIME, NPTIME, PSPACE, NPSPACE víte, že platí ?

Uveďte příklad (ANO/NE) problému, jenž je rozhodnutelný, ale který nepatří do NPSPACE:

Název:

Vstup:

Otázka:

---

[9] **(12 bodů)**

Okruh: Rozhodnutelné a částečně rozhodnutelné problémy. Rekurzivní a rekurzivně spočetné množiny (jazyky). Postova věta. Rekurzivní a částečně rekurzivní funkce. Church Turingova teze. Univerzální Turingův stroj.

Příklad:

Univerzální Turingův stroj U očekává vstup ve tvaru $Kod(M) \cdot w$; pro něj pak (krok po kroku) simuluje činnost stroje M na vstupu w. Během této simulace si pochopitelně musí neustále "pamatovat" aktuální stav řídicí jednotky stroje M. Jak je toto u stroje U realizováno ?

---

[10] **(12 bodů)**

Okruh: Nerozhodnutelnost Halting problému. Převeditelnost mezi problémy. Další nerozhodnutelné problémy. Riceova věta; její aplikace.

Příklad:

Vyznačte (pokud možno, i stručně zdůvodněte), pro které z následujících problémů plyne/neplyne nerozhodnutelnost z Riceovy věty.

Instancí (tj. vstupem problému) je vždy Turingův stroj M, proto uvádíme jen otázky:

a/ Zastaví se M na řetězec 001 ?

b/ Má M více než sto stavů ?

c/ Má v nějakém případě výpočet stroje M více kroků než tisícinásobek délky vstupu ?

d/ Platí, že pro libovolné n se M na vstupech délky nejvýše n vícekrát zastaví než nezastaví ?

e/ Je pravda, že pro lib. vstupní slovo M realizuje jeho zdvojení ?