EXERCÍCIOS: 1

RAUL H.C. LOPES

1. Introdução

A seguir você encontra uma série de exercícios que você pode considerar como "treino coletivo" para o grande e definitivo jogo que se aproxima. Você encontrará aqui exercícios sobre os fundamentos de análise e correção de algoritmos e classes de complexidade. Importante: não existe (como não poderia haver) a pretensão de cobrir tudo o que foi apresentado. Por outro lado, considero que muitos destes exercícios você já resolveu como parte do seu estudo diário ao longo do semestre: é até desrespeitoso em relação à instituição em que você estuda e, portanto, em relação a você mesmo assumir que você vá estudar a matéria toda nas próximas duas semanas. Em relação ao grau de dificuldade, lembre-se: você vai ser um mestre em Ciência da Computação e os alunos de Ciência da Computação estudam exatamente esses tópico em um curso de um semestre e fazem os mesmos trabalhos práticos que você está fazendo. (Well... pelo menos, quando eu dou a disciplina.) O que não quer dizer que esta disciplina seja trivial: é difícil. Mas, não impossível.

Aliás... Nossa prova fica marcada para o dia 15/04. O horário? Vocês determinam. Mas, se demorarem para decidir, eu determinarei. Infelizmente, compromissos urgentes vão aparecendo.

BTW (ou aliás aliás?), eu não dou mais aulas este semestre.

1.1. **Notação.** Assuma que:

- $\mathcal{P}$, $\mathcal{P}$, $\mathcal{NP}$-complete, $\mathcal{NP}$-hard, co-$\mathcal{NP}$ denotam as classes de problemas.
- TM abrevia Máquina de Turing determinística.
- NTM abrevia Máquina de Turing não determinística.
- L *é* Turing accpetable significa que existe TM que aceita L.

2. Ordenação

Questão 1. *Em relação à ordenação de seqüências apresente:*

(1) *condições de correção para um método de ordenação.*

(2) *algoritmos com argumento de correção para cada método de ordenação estudado.*
(3) *estudo completo de complexidade para os algoritmos de cada método de ordenação, abrangendo: melhor, pior caso e caso médio, com respectivas relações de recorrência, desenvolvimento e fórmula fechada.*
(4) *distinção entre ordenação* in-locus *e ordenação com construção de nova seqüência.*
(5) *conceito de ordenação estável.*

3. Indexação

Questão 2. (1) *Apresente algoritmos para criação de e pesquisa em árvores binárias e árvores B.*
(2) *Defina condições de correção para os algoritmos acima.*
(3) *Prove correção dos algoritmos acima.*
(4) *Estude pior caso e melhor caso de pesquisa para as duas estruturas.*
(5) *Apresente ao menos duas representações diferentes para nós de árvores B e estude seus impactos na complexidade de pesquisa.*

Questão 3. *Prove todos os teoremas propostos em sala de aula para a árvore de sufixos Ukkonen.*

4. Complexidade de problemas exponenciais

Questão 4. *Desenvolva completamente o algoritmo de Kruskal para cálculo árvore geradora mínima. Estabeleça os critérios de sua correção. Prove sua correção. Estude sua complexidade.*

Questão 5. *Apresente e prove correção de algoritmo para TSP euclideano que gere caminho com peso equivalente a no máximo duas vezes o peso do melhor caminho.*

5. Máquinas de Turing

Questão 6. (1) *Defina formalmente* TM.
(2) *Defina formalmente* NTM.
(3) *Proponha* TM *que aceite a linguagem* 0^n1^n.
(4) *Proponha* TM *que aceite a linguagem* ww^R, *onde* w^R *denota o reverso de* w.
(5) *Defina linguagem recursiva.*
(6) *Defina linguagem recursivamente enumerável.*

Questão 7. *Prove:*

(1) *Se* L *é recursiva então* $\bar{L}$ *também é.*
(2) *Se* L *é a linguagem de uma* TM M *e* M *tem tamanho polinomial, então* $\bar{L}$ *é a linguagem de uma* TM *de tamanho polinomial.*

Questão 8. *Apresente, justificando formalmente, uma linguagem que não é recusivamente enumerável.*

Questão 9. *Defina as seguintes classes de problema:*

(1) *Classe* $\mathcal{P}$.
(2) *Classe* $\mathcal{P}$.
(3) *Classe* $\mathcal{P}$.
(4) *Classe* $\mathcal{NP}$*-complete.*
(5) *Classe* $\mathcal{NP}$*-hard.*
(6) *Classe co-*$\mathcal{NP}$.

Questão 10. *Prove que o complemento de qualquer linguagem contida em* $\mathcal{P}$ *está em* $\mathcal{P}$.

Questão 11. *Seja*

$$L_0.L_1 = \{uv \mid u \in L_0 \wedge v \in L_1\}$$

one uv *é a concatenação de* u *com* v*. Prove que se* $L_0, L_1 \in \mathcal{P}$ *então* $L_0.L_1 \in \mathcal{P}$.

Questão 12. *Seja* $\boldsymbol{CSAT}$ *o problema da satisfatibilidade de fórmulas do cálculo proposicional construídas usando apenas variáveis proposicionais, conjunção, disjunção e negação, tais que a negação só é usada para negar variaveis proposicionais. Assumindo que* $\boldsymbol{SAT}$ *é* $\mathcal{NP}$*-complete, prove que* $\boldsymbol{CSAT}$ *também é.*