

Instituto de Computação – UNICAMP
Complexidade de Algoritmos I – Turma A
Exercícios: **Técnicas de projeto de algoritmos avançadas**

- Os exercícios devem ser submetidos como um arquivo em formato PDF (digitado ou manuscrito digitalizado), no prazo estipulado, na página <http://www.ic.unicamp.br/~lehilton/mo417a/submit/>.
- Só serão aceitas listas com todas questões respondidas, mas serão corrigidos **apenas** os itens sorteados em <http://www.randomresult.com/ticket.php?t=261735JDW88>.

Questão 1. (CLRS) Considere o problema de imprimir um parágrafo com perfeição usando uma fonte monoespaçada (todos os caracteres têm a mesma largura). O texto de entrada é uma sequência de n palavras de comprimentos l_1, l_2, \dots, l_n , medidos em caracteres. Queremos imprimir este parágrafo de forma organizada em várias linhas que contenham no máximo M caracteres cada. Nosso critério de “organização” é o seguinte. Se uma determinada linha contém palavras de i a j , onde $i \leq j$ e deixamos exatamente um espaço entre as palavras, o número de espaços a mais no final da linha é $M - j + i - \sum_{k=i}^j l_k$, que deve ser não negativo para que as palavras se encaixem na linha. Desejamos minimizar a soma sobre todas as linhas com exceção da última dos cubos dos números de espaço a mais nas extremidades das linhas. Dê um algoritmo baseado em programação dinâmica para imprimir um parágrafo de n palavras. Analise o tempo de execução e os requisitos de espaço do seu algoritmo.

Questão 2. (CLRS) Suponha que em um problema de mochila 0-1, a ordem dos itens quando classificados por aumento de peso é a mesma ordem que quando classificados por valor decrescente. Dê um algoritmo eficiente para encontrar uma solução ótima para essa variante do problema de mochila e argumente que seu algoritmo está correto.

Questão 3. (CLRS) Generalize o algoritmo de Huffman para palavras de código ternárias (i.e., palavras de código usando os símbolos 0, 1 e 2) e prove que ele produz códigos ternários ótimos.