MCTA003-17 - ANÁLISE DE ALGORITMOS
CENTRO DE MATEMÁTICA, COMPUTAÇÃO E COGNIÇÃO
UNIVERSIDADE FEDERAL DO ABC
PROF. MAYCON SAMBINELLI

Algoritmos Gulosos

1. No problema da Mochila Inteira são dados n itens, cada item i tendo um valor v_i e um peso w_i, e é dada uma capacidade W da mochila; queremos selecionar um subconjunto $S \subseteq \{1, 2, \ldots, n\}$ dos itens cujo peso $\sum_{i \in S} w_i$ seja no máximo W e cujo valor $\sum_{i \in S} v_i$ seja máximo. A estratégia gulosa de escolher sempre o item com maior razão $valor/peso$ encontra uma solução ótima para o problema da Mochila Inteira? Justifique.

2. Considere novamente o problema de seleção de tarefas compatíveis: dado um conjunto $T = \{t_1, \ldots, t_n\}$ com n tarefas onde cada $t_i \in T$ tem um tempo inicial s_i e um tempo final f_i, encontre o maior subconjunto de tarefas mutuamente compatíveis. Considere o seguinte algoritmo guloso: enquanto houver tarefas compatíveis com as já escolhidas, escolha a tarefa que começa por último (com maior valor s_i) que seja compatível com as tarefas já escolhidas. Prove que essa estratégia resulta em um algoritmo ótimo.

3. PROBLEMA DO CONJUNTO INDEPENDENTE MÁXIMO EM FLORESTAS
Entrada: uma F uma floresta (um grafo no qual toda componente conexa é uma árvore).
Saída: o maior tamanho de um conjunto independente em F, que denotamos por $\alpha(F)$.

(a) Seja F uma floresta e seja S um conjunto independente máximo de F. Prove que se $u \in S$, então $S \setminus \{u\}$ é um conjunto independente máximo de $F - N_F[u]$, onde $N_F[u] = N_F(u) \cup \{u\}$.

(b) Prove que se $u \in F$ tal que $d_F(u) \leq 1$ e $S^{'}$ é um conjunto independente máximo de $F - N_F[u]$, então $S^{'} \cup \{u\}$ é um conjunto independente máximo de F.

(c) Apresente um algoritmo guloso para o Problema do Conjunto Independente Máximo em Florestas.

(d) Analise o algoritmo que você apresentou no item anterior.

4. Um *emparelhamento* em um grafo G é um conjunto de arestas $F \subseteq E(G)$ tal que nenhum par de arestas em F possui vértices em comum. Note que uma aresta $e \in E(G)$ sozinha já é um emparelhamento. É interessante, portanto, encontrar um emparelhamento de tamanho maior possível. Descreva um algoritmo guloso que devolve um emparelhamento em um dado grafo G. Seu algoritmo é ótimo?