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

Algoritmos Gulosos

INSTRUÇÕES

Para qualquer exercício dessa lista que lhe peça para fornecer um algoritmo/solução para um problema, a menos que explicitamente dito o contrário, você deve:

- descrever em palavras qual é a **ideia** do seu algoritmo. Não é para escrever em prosa o que está escrito no pseudo-código, explique a ideia;
- escrever um pseudocódigo (pode ser em alto nível);
- provar ou fornecer um argumento intuitivo para sua correção. É claro que, se o exercício explicitamente pede uma prova, então um argumento intuitivo não será o suficiente.
  - Observe que em problemas de otimização, duas coisas são importantes:
    * argumentar que seu algoritmo devolve sempre uma solução viável;
    * argumentar se o algoritmo é ótimo ou não.
- analisar o tempo do seu algoritmo.

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. Sejam n e m dois inteiros positivos. Seja $\mathcal{C} = \{C_1, \ldots, C_m\}$ uma coleção com m conjuntos de números entre 1 e n, isto é, $C_i \subseteq \{1, \ldots, n\}$ para cada $1 \leq i \leq m$.

Uma *cobertura* $\mathcal{S}$ é um subconjunto de $\mathcal{C}$ cuja união é igual a $\{1, \ldots, n\}$, isto é, $\mathcal{S} \subseteq \mathcal{C}$ é uma cobertura se $\bigcup_{C_i \in \mathcal{S}} C_i = \{1, \ldots, n\}$. O tamanho de uma cobertura $\mathcal{S}$ é sua cardinalidade, $|\mathcal{S}|$. Uma *cobertura mínima* é uma cobertura de menor tamanho possível.

O problema da *Cobertura de Conjuntos* consiste em, dados n, m e $\mathcal{C} = \{C_1, \ldots, C_m\}$, com cada $C_i \subseteq \{1, \ldots, n\}$, encontrar uma cobertura de cardinalidade mínima. Descreva um algoritmo guloso para o problema da cobertura de conjuntos. Você pode assumir que $\bigcup_{i=1}^{m} C_i =$

$\{1, \ldots, n\}$, isto é, existe ao menos uma solução viável na instância de entrada. O seu algoritmo é ótimo? Se sim, demonstre; caso contrário, apresente um contraexemplo.

4. Nesta questão, considere o problema de fazer troco para n centavos usando o menor número total de moedas. Você pode assumir que existe um número infinito de moedas disponíveis para cada valor.

(a) Forneça um algoritmo guloso ótimo para fazer troco tendo disponíveis moedas de 1, 5, 10, 25 e 50 centavos. *Dica: para provar que seu algoritmo é ótimo, assuma que ele não é. Qual é a diferença de uma solução ótima para a solução do seu algoritmo?*

(b) O algoritmo que você deu acima continua ótimo se o conjunto de moedas disponíveis for 1, 7, 10 e 16? Justifique.

5. Um *grafo caminho* G é um grafo que consiste apenas de um caminho, isto é, $V(G) = \{v_1, v_2, \ldots, v_n\}$ e $E(G) = \{v_i v_{i+1} \colon 1 \leq i < n\}$ e seja $w \colon V(G) \to \mathbb{R}^+$ uma função de peso sobre **os vértices** de G. Assim, para qualquer conjunto $X \subseteq V(G)$ de vértices, o peso de X, $w(X)$, é naturalmente definido como $w(X) = \sum_{v \in X} w(v)$.

Um conjunto S de vértices de um grafo qualquer é *independente* se para quaisquer dois vértices $u, v \in S$, não existe aresta uv no grafo.

O problema do *Conjunto Independente de Peso Máximo* consiste em receber um grafo caminho G e uma função w de peso nos vértices e o objetivo é encontrar um subconjunto S de vértices tal que S é um conjunto independente e $\sum_{v \in S} w(v)$ é máximo. Responda:

(a) Explique por que o algoritmo a seguir é guloso e não é ótimo.

```
1: seja S = ∅
2: Enquanto V(G) ≠ ∅ faça
3:     escolha v ∈ V(G) com peso w(v) máximo
4:     S = S ∪ {v}
5:     remova v e seus vizinhos de G
6: Devolve S
```

(b) Explique por que o algoritmo a seguir é guloso e não é ótimo.

```
1: seja S_1 = {v_i : i é ímpar}
2: seja S_2 = {v_i : i é par}
3: Devolve o conjunto de maior peso dentre S_1 e S_2
```

6. 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?