

Análise e Técnicas de Algoritmos

Lista de Exercícios - Métodos Gulosos

Aluno(a):

1. O carro da universidade faz frequentemente o trajeto Campina Grande-João Pessoa, utilizando a BR-230. Se o tanque de gasolina do carro está cheio, tem combustível suficiente para percorrer k quilômetros. Existe um mapa detalhado com as distâncias entre postos de gasolina ao longo da BR-230. Sejam $d_1 < d_2 < \dots < d_n$ as coordenadas de todos os n postos ao longo do citado trajeto, onde d_i é a distância de Campina Grande até o posto de gasolina p_i . Ainda é conhecido que a distância entre postos vizinhos não ultrapassa k quilômetros. O objetivo é sair de Campina Grande e chegar em João Pessoa com o menor número possível de paradas para abastecimento.
 - (a) Escreva, em pseudo-código, um algoritmo que utiliza a estratégia gulosa e resolve o problema descrito.
 - (b) Prove que o seu algoritmo sempre produz a solução ótima.
 - (c) Determine a complexidade do algoritmo em função de n .
2. Descreva formalmente o problema da mochila fracionária. Defina uma solução gulosa para este problema. Esta solução deve ser baseada na solução genérica que vimos em sala de aula. Destaque cada um dos elementos da solução genérica na solução gulosa do problema da mochila.
3. Os moradores da cidade de Lamaceira sempre desejaram ter suas ruas pavimentadas, mas não implementaram a idéia por custar muito caro. Entretanto, depois de muitos anos de intenso inverno, eles decidiram não mais ter que sujar os pés de lama. Como tinham pouco dinheiro, concordaram em pavimentar apenas o bastante para garantir que entre dois pontos sempre é possível utilizar um caminho pavimentado. Os moradores de Lamaceira não se importam de andar muito para economizar dinheiro.
 - (a) Como resolver o problema dos moradores da cidade de Lamaceira?
 - (b) Ilustre a sua solução com um exemplo.

4. O método guloso sempre fornece uma solução ótima? Justifique a sua resposta através de um exemplo.
5. Sobre métodos gulosos:
- Apresente sua definição.
 - Quais as suas vantagens?
 - Quais as suas desvantagens?
6. Considere o seguinte problema de coleta de cupons. Existe uma certa quantidade de diferentes tipos de caixa de biscoito. Em cada caixa de biscoito encontramos um cupom que dá um desconto na compra de uma outra caixa de biscoito (não necessariamente do mesmo tipo). É possível utilizar múltiplos cupons para comprar uma nova caixa de biscoito, até o valor de obtê-la grátis. Não é possível receber dinheiro de volta, mesmo se sobrar cupom. É necessário comprar uma caixa de cada tipo de biscoito, gastando o menor valor possível. Descreva um algoritmo eficiente, que recebe como entrada, para cada tipo de biscoito, seu preço, o valor do cupom e a marca contemplada no cupom, e retorna a ordem ótima de compra das caixas de biscoito.
7. Considere um arquivo texto onde encontramos apenas os seguintes caracteres: a(25), b(12), c(4), d(4), e(7), i(30), z(18). Entre parênteses está indicada a frequência de ocorrência de cada caractere. Aplicar a estratégia de Huffman para a compressão de arquivos. Qual seria o fator de compressão neste caso?
8. Considere uma lista ordenada de n palavras. O tamanho da i -ésima palavra é w_i . Para simplificar nosso problema, vamos assumir que não existe espaço entre palavras. O objetivo do problema é quebrar a lista ordenada de palavras em linhas. Não é possível reordenar as palavras. O tamanho de uma linha é então a soma dos tamanhos das palavras daquela linha. A linha ideal tem tamanho L . Nenhuma linha pode ser maior do que L , embora possa ser menor. Uma multa para uma linha de tamanho K é $L - K$. A multa total é a **soma** das multas das linhas. O problema é resolvido se a escolha das quebras produzir uma multa total que é mínima.
- ```

for $i \leftarrow 1$ to n do
 if i -ésima palavra couber na linha corrente then
 coloque a i -ésima palavra nesta linha
 else
 coloque a i -ésima palavra em uma nova linha
 end if
end for

```
- O algoritmo guloso acima resolve o problema? Justifique a sua resposta.