

1ª. ETAPA

1. Escreva um algoritmo que obtenha o código de Huffman ternário, ou seja, um código de Huffman em que podemos codificar os símbolos com 0,1 e 2. Exemplifique esse algoritmo e também o código de Huffman normal para o seguinte exemplo: um texto com 100.000 caracteres onde os símbolos são A, B, C, D, E, F e G com frequências 40%, 15%, 11%, 10%, 12%, 7% e 5%. Quantos bits esse texto codificado terá tanto no caso normal como no ternário? Quantos bits esse texto teria se usássemos uma codificação de tamanho fixo? Compare cada algoritmo: melhorou ou piorou?
2. Considere um conjunto de livros enumerados de 1 a n . Suponha que o livro i tem peso p_i e que $0 < p_i < 1$ para cada i . Escreva um algoritmo para acondicionar os livros no menor número possível de envelopes de modo que cada envelope tenha no máximo dois livros e o peso do conteúdo de cada envelope seja no máximo 1. Prove a corretude do seu algoritmo.
3. Seja $1, \dots, n$ um conjunto de tarefas. Cada tarefa consome um dia de trabalho; durante um dia somente uma das tarefas pode ser executada. Os dias são numerados de 1 a n . A cada tarefa T está associado um prazo P_T : a tarefa deve ser executada em algum dia do intervalo $1, \dots, P_T$. A cada tarefa T está associada uma multa $M_T \geq 0$. Se uma tarefa T é executada depois do prazo P_T , sou obrigado a pagar a multa M_T (mas a multa não depende do número de dias de atraso). Escreva um algoritmo guloso para programar as tarefas (ou seja, informar em qual dia cada tarefa será realizada) de modo a minimizar a multa total. Prove a corretude e analise o consumo de tempo.
4. O problema de seleção de atividades é o problema de encontrar, a dada coleção de intervalos, uma subcoleção de tamanho máximo de intervalos dois a dois disjuntos. Nem todo algoritmo guloso resolve esse problema. Mostre que nenhuma das três ideias a seguir resolve o problema.
Ideia 1: Escolha a atividade de menor duração dentre as que são compatíveis com as atividades já escolhidas.
Ideia 2: Escolha uma atividade que seja compatível com as já escolhidas e intercepta o menor número possível de atividades ainda não escolhidas.
Ideia 3: Escolha a atividade compatível com as já selecionadas que tenha o menor instante de início.
5. Um servidor tem n usuários esperando para serem servidos. O tempo de serviço requerido por usuário é conhecido previamente: é t_i minutos para o usuário i . Portanto se, por exemplo, os usuários são servidos em ordem crescente de i , então o i -ésimo usuário tem de esperar $\sum_{j=1}^i t_j$ minutos.
Queremos minimizar o tempo total de espera

$$T = \sum_{i=1}^n (\text{tempo gasto pelo usuário } i \text{ na espera}).$$

Forneça um algoritmo guloso para calcular a ordem ótima na qual devemos processar os usuários. Explique resumidamente propriedade da escolha gulosa.

6. Quero dirigir um carro de uma cidade A a uma cidade B ao longo de uma rodovia. O tanque de combustível do carro tem capacidade suficiente para cobrir n quilômetros. O mapa da rodovia indica a localização dos postos de combustível. Escreva um algoritmo que garanta uma viagem com o número mínimo de reabastecimentos. Explique resumidamente a propriedade de escolha gulosa do seu algoritmo.
7. Dado um vetor de n dígitos (cada dígito com valor de 0 a 9), encontre a menor soma possível de dois números formados por dígitos desse vetor. Todos os dígitos devem ser usados para formar os dois números. Cada dígito só pode ser usado uma vez. Esse problema tem a propriedade da escolha gulosa? Se sim, prove a propriedade da escolha gulosa e escreva um algoritmo guloso para resolvê-lo. Se não, elabore um algoritmo de programação dinâmica que resolva o problema.

8. Os dentistas são extremamente meticulosos em seu trabalho, tendo que agir com muita precisão em todas as suas atividades. Pedro é um dentista meticoloso como todos os outros. Infelizmente sua secretária não é muito organizada e, com o intuito de ajudar sempre os pacientes, aceita que eles marquem consultas no horário que quiserem, sem se preocupar com os demais horários marcados, ocasionando vários conflitos de horários que muito incomodaram Pedro e os pacientes. Por exemplo, se uma consulta começar às 9 horas e durar 2 horas, nenhuma outra consulta deveria ser marcada para iniciar às 10 horas. Ao perceber que sua agenda tinha conflito de horários, Pedro pediu sua ajuda para descobrir a maior quantidade de consultas que podem ser atendidas sem sobreposição.

Tarefa: Você deve escrever um algoritmo guloso que, dados os horários de início e término das consultas agendadas pela secretária, responda a quantidade máxima de consultas que podem ser atendidas sem sobreposição. Você recebe como entrada a quantidade de consultas agendadas e os horários de início e fim de cada consulta agendada.

Explique a propriedade da escolha gulosa.

```
if(c_atual(finish) > c_selecionada(start)) return c_atual
```