

Construção e Análise de Algoritmos
Lista de exercícios 3
Algoritmos Gulosos e Complexidade Computacional

1. Escreva um algoritmo eficiente que receba como entrada um conjunto de variáveis $x_1, \dots, x_n$ e dois conjuntos I e D de pares (x_i, x_j) de variáveis. Os pares (x_i, x_j) em I representam restrições de igualdade $x_i = x_j$ e os pares (x_a, x_b) em D representam restrições de desigualdade $x_a \neq x_b$. Seu algoritmo deve responder se é possível ou não satisfazer todas as restrições em I e em D . Por exemplo, a seguinte entrada não é satisfatória: $I = \{(x_1, x_2), (x_2, x_3), (x_3, x_4)\}$ e $D = \{(x_1, x_4)\}$.

2. Um certo Fulano deseja fazer uma festa e está decidindo quem deve chamar. Ele tem n amigos para convidar e tem uma lista dos pares de amigos que se conhecem. Ele quer que ninguém se sinta deslocado, mas também quer que a festa seja interessante e que pessoas que não se conhecem façam amizade. Assim ele se colocou as seguintes restrições: Para cada convidado, devem existir pelo menos dez pessoas na festa que ele conhece e dez pessoas na festa que ele não conhece. Faça um algoritmo que receba uma lista com n pessoas e uma lista com os pares dessas pessoas que se conhecem e devolva o maior número de pessoas que poderão ser convidadas sob estas restrições. (a) Descreva com palavras como será o seu algoritmo. (b) Escreva o algoritmo em pseudo-código.

3. Seja $1, \dots, n$ um conjunto de tarefas. Cada tarefa consome um dia de trabalho; durante um dia de trabalho somente uma das tarefas pode ser executada. Os dias de trabalho são numerados de 1 a n . A cada tarefa T está associado um prazo P_T : a tarefa deveria ser executada em algum dia do intervalo $1, \dots, P_T$. A cada tarefa T está associada uma multa não-negativa M_T . Se uma dada 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). Problema: Programar as tarefas (ou seja, estabelecer uma bijeção entre as tarefas e os dias de trabalho) de modo a minimizar a multa total. Escreva um algoritmo guloso para resolver o problema. Prove que seu algoritmo está correto. Analise o consumo de tempo.

4.

(a) Explique informalmente o que significa um problema ser NP-Completo. Do ponto de vista prático, qual é a relevância de se determinar que um certo problema é NP-Completo?

(b) Para cada uma das afirmações abaixo, diga se ela é *verdadeira*, *falsa*, *verdadeira se $P \neq NP$* ou *falsa se $P \neq NP$* . Dê uma justificativa curta para cada resposta.

1. Não há problemas em P que são NP-Completo
2. Existe apenas algoritmo exponencial para o problema da parada
3. Existem problemas em P que estão em NP
4. Existem problemas em NP que não estão em P
5. Se A pode ser polinomialmente reduzido a B, e B é NP-Completo, então A é NP-Completo
6. Se A pode ser polinomialmente reduzido a B, e $B \in P$, então $A \in P$
7. O problema de encontrar a rota mínima do Caixeiro Viajante é NP-Completo

5. Dado um grafo G , uma coloração é uma atribuição de cores a seus vértices de forma que vértices adjacentes tenham cores diferentes. Seja 3CORES o problema de decidir se, dado um grafo G como entrada, G pode ser colorido com 3 cores. Mostre que 3CORES é NP-Completo.



Figura 1: Use essas engrenagens na questão 1

6. Dizemos que um grafo G está parcialmente rotulado se alguns de seus vértices possuem um número inteiro como rótulo. Dado um vértice rotulado v de G , seja $r(v)$ o seu rótulo. Seja CAMPO-MINADO o problema de decidir se, dado como entrada um grafo G parcialmente rotulado, G pode ser completamente rotulado de forma que qualquer vértice v com rótulo positivo tenha exatamente $r(v)$ vizinhos com rótulo negativo. Prove que CAMPO-MINADO é NP-Completo. Dica: Imagine rótulos negativos como bombas do campo minado. Tente reduzir 3SAT para este problema: cada variável tendo dois vértices no grafo com um vizinho comum com rótulo igual a 1. Pense nas bombas do campo minado como uma atribuição de Verdadeiro.

7. Seja DOMINANTE o problema de decidir se, dado como entrada um grafo G e um inteiro $k > 0$, existe um conjunto D com k vértices de G tal que todo vértice de G está em D ou é adjacente a algum vértice de D .

- Prove que DOMINANTE é NP-Completo usando o problema 3SAT
- Prove que DOMINANTE é NP-Completo usando o problema COB-VERT da Cobertura de vértices.

8. Seja MOCHILA o problema de decidir se, dados inteiros positivos P e V e dado um conjunto S onde cada elemento $s \in S$ possui um peso $p(s)$ e um valor $v(s)$, existe um subconjunto S' de S tal que a soma dos pesos dos elementos de S' seja menor ou igual a P e a soma dos valores dos elementos de S' seja maior ou igual a V . Prove que MOCHILA é NP-Completo. **Dica:** Problema SOMA-SUBC (ou SUBSET-SUM) da soma de subconjuntos.