MCTA003-17 -- Análise de Algoritmos

Prof. Ronaldo C. Prati

lista compilada pela Profa. Carla Negri Lintzmayer

Lista 4

1. Execute o algoritmo de programação dinâmica para o problema da Mochila Inteira e mostre a matriz final preenchida por ele sobre a seguinte entrada:
   $W = 10$, $v_1 = 15$, $w_1 = 3$, $v_2 = 12$, $w_2 = 4$, $v_3 = 17$, $w_3 = 2$, $v_4 = 18$, $w_4 = 5$, $v_5 = 15$, $w_5 = 2$, $v_6 = 10$, $w_6 = 3$, $v_7 = 15$, $w_7 = 1$.
   - Explique como você preencheu a entrada da linha referente ao quarto item e à capacidade 9.
2. Dada a matriz totalmente preenchida pelo algoritmo de programação dinâmica para o problema do Alinhamento de Sequências, mostre como construir um alinhamento para as sequências da entrada.
3. Considere o seguinte problema.
   Seja $G = (V, E)$ um grafo que constitui apenas de um caminho, isto é, $V(G) = \{v_1, v_2, \ldots, v_n\}$ e $E(G) = \{v_i v_{i+1} : 1 \leq i < n\}$ e seja $w : V(G) \to \mathcal{R}$ uma função de peso sobre \textbf{os vértices}.
   O problema do **Conjunto Independente de Peso Máximo** consiste em receber um grafo caminho G e a função w de peso nos vértices e o objetivo é encontrar um subconjunto S de vértices tal que S é um conjunto independente (isto é, para quaisquer dois vértices $u, v \in S$, não existe a aresta uv em G) e $\sum_{v \in S} w(v)$ é máximo.
   (a) Mostre que o algoritmo a seguir nem sempre encontra uma solução ótima para o problema.

   1: seja $S = \emptyset$
   2: **while** $V(G) \neq \emptyset$ **do**
   3: escolha $v \in V(G)$ com $w(v)$ máximo
   4: $S = S \cup \{v\}$
   5: remova v e seus vizinhos de G

   (b) Mostre que o algoritmo a seguir também nem sempre encontra uma solução ótima para o problema.

   1: seja $S_1 = \{v_i : i$ é ímpar $\}$
   2: seja $S_2 = \{v_i : i$ é par $\}$
   3: retorne o conjunto de maior custo dentre S_1 e S_2

(c) Escreva e analise um algoritmo de programação dinâmica que resolve esse problema otimamente.

4. O algoritmo *Floyd-Warshall* consegue executar e terminar mesmo se o grafo da entrada possuir um ciclo negativo.
   Porém, ele não executa *corretamente*.
   Se para algum i, com $1 \leq i \leq |V(G)|$, tivermos $W[i][i] < 0$ na matriz W construída pelo algoritmo após a última iteração, então é garantido que existe um ciclo negativo.
   Por quê?
5. Defina: algoritmo eficiente, problema de decisão, problema de otimização, certificado, algoritmo verificador, classes P, NP, NP-completo e NP-difícil.
6. É verdade que se reduzirmos um problema em P para um problema em NP, então P $=$ NP? Justifique.
7. Seja A um problema NP-completo. Seja B um outro problema qualquer.
   Diga o que podemos concluir ao reduzir A para B e ao reduzir B para A.
8. Argumente por que não podemos dizer que um problema de otimização é NP-completo.
9. Mostre que o problema Set Cover é NP-completo usando o problema Vertex Cover, que é NP-completo.

> PROBLEMA: TSP
> ENTRADA: grafo G, função de custo c nas arestas.
> OBJETIVO: ciclo hamiltoniano cuja soma dos custos das arestas é mínima.

> PROBLEMA: TSPK
> ENTRADA: grafo G, função de custo c nas arestas, valor k.
> DECISÃO: existe ciclo hamiltoniano cuja soma dos custos das arestas menor ou igual a k?

> PROBLEMA: SET COVER
> ENTRADA: conjunto $T = \{u_1, \ldots, u_n\}$ de n elementos, coleção S_1,...,S_m de subconjuntos de T ($S_i \subseteq T$ para todo i), inteiro k. DECISÃO: existem no máximo k conjuntos S_i tal que a união deles é T?

> PROBLEMA: VERTEX COVER
> ENTRADA: grafo G e inteiro ℓ. DECISÃO: existe subconjunto de vértices $S \subseteq V(G)$ tal que $|S| \leq \ell$ e para toda aresta $uv \in E(G)$ vale que pelo menos um dentre u e v estão em S?

10. Mostre que se o TSPk pode ser resolvido em tempo polinomial então o TSP também pode.
11. Faça um algoritmo de programação dinâmica para calcular $\binom{n}{k}$ sem usar a fórmula fechada para isso.
    Por definição, $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ para $1 \leq k < n$ e $\binom{n}{n} = \binom{n}{0} = 1$.