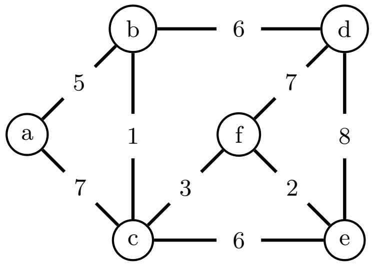
Universidade Federal da Paraíba
Centro de Informática
Disciplina: Análise e Projeto de Algoritmos
Professor: Bruno Bruck

Lista de exercícios (Unidade 2)

1) Considere o grafo G = (V, A) dado a seguir:



a) Explique o algoritmo de **Kruskal** utilizando o grafo acima para ilustrar o **passo a passo** do algoritmo.

b) Em termos de eficiência de tempo de execução, para a implementação do algoritmo de Kruskal é mais indicado utilizar listas de adjacência ou matriz de adjacência? Justifique sua resposta.

2) Considere o seguinte pseudocódigo do algoritmo de Prim:

```
function Prim(G, w, r)
    chave = []
    pai = []
    for each i ∈ V do
        chave[i] = +∞
        pai[i] = null
    chave[r] = 0
    Q = criarLista(V, chave)
    while Q ≠ ∅ do
        u = acharMinimo(Q)
        for each v ∈ adj[u] do
            if v ∈ Q and w[u][v] < chave[v] then
                pai[v] = u
                chave[v] = w[u][v]
                atualizar(Q)
    return (chave, pai)
```

a) Qual a complexidade do algoritmo de Prim utilizando-se listas de adjacência para a representação de G e uma fila de prioridade baseada em heap binário para a implementação de Q? Justifique sua resposta.

b) Explique por que a representação do grafo G por meio de uma matriz de adjacência faz com que o algoritmo tenha complexidade $\Theta(|V|^2)$.

c) Como podemos modificar o algoritmo de Prim para resolver o problema do Caminho Mínimo de origem única?

3) Considere uma instância do Problema da Mochila 0-1 com capacidade $M = 7$ e um conjunto de 4 itens com os vetores $p = [3, 5, 2, 4]$ e $v = [4, 5, 7, 8]$ de pesos e valores, respectivamente.

a) Explique como são criados os subproblemas para o algoritmo de Programação Dinâmica do Problema da Mochila.

b) Resolva essa instância do Problema da Mochila 0-1 usando a técnica de Programação Dinâmica *bottom-up* ensinada em sala de aula.

c) Explique qual o procedimento que deve ser realizado para recuperar a solução ótima da tabela utilizando a solução encontrada no item anterior como exemplo.

4) Considere a seguinte definição de um problema de otimização:

> "Seja $G = (V, A)$ um grafo onde, V é o conjunto de vértices que representa os convidados de uma festa, e A é o conjunto de arcos do grafo. Um arco $(i, j) \in A$ representa uma relação de afinidade entre os convidados i e j e possui um certo peso c_{ij}. Os pesos c_{ij} podem assumir tanto valores positivos quanto negativos. No caso de valores positivos, quanto maior, melhor é a afinidade entre os respectivos convidados. Casos onde o valor é negativo representam situações onde é prejudicial designar os convidados i e j para a mesma mesa. Além disso, existe à disposição um conjunto P de mesas. Cada mesa $p \in P$ deve conter no mínimo L_p convidados e no máximo U_p. O objetivo do problema é distribuir os convidados nas mesas disponíveis de forma a maximizar a afinidade total."

a) Desenvolva um algoritmo guloso para prover uma solução viável para esse problema. Esse algoritmo deve ser apresentado em formato de pseudocódigo e acompanhado por uma descrição geral da ideia do algoritmo.

b) Faça a análise de complexidade do algoritmo e obtenha um limite assintoticamente restrito.

5) **Discorra** sobre a veracidade das seguintes afirmações fornecendo argumentos consistentes para embasar sua resposta.

a) Um dado problema de otimização possui um algoritmo de força bruta que roda em tempo exponencial, mas foi observado que é possível utilizar a técnica de Programação Dinâmica para tal problema. Mesmo assim não há garantias de que o algoritmo de Programação Dinâmica rode em tempo polinomial;

b) Quanto maior o nível de superposição de subproblemas, melhor tende a ser a eficiência do algoritmo de Programação Dinâmica em comparação com o algoritmo de Força Bruta;

c) Qualquer problema de otimização pode ser resolvido utilizando Programação Dinâmica, mas nem sempre o resultado vai ser um algoritmo eficiente.

6) Considere o seguinte problema de otimização, conhecido como **Problema de Coloração de Grafos (PCG)**:

Problema de Coloração de Grafos
Entrada: Grafo simples e não orientado $G = (V, E)$, com conjunto de vértices V e conjunto de arestas E.
Tarefa: Atribuir cores ao vértices de G de modo que dois vértices vizinhos possuam cores distintas. O objetivo é utilizar o menor número de cores.

Observe que uma solução trivial é utilizar $|V|$ cores distintas, mas o que queremos é utilizar o menor número de cores possível. Na sua resposta, considere que as cores disponíveis são representadas pelos inteiros $1, 2, \ldots, |V|$. A Figura 1 ilustra o problema da coloração.

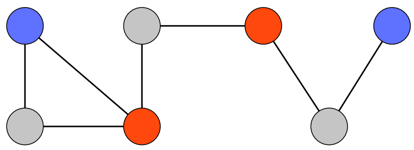


Figure 1: Um grafo que pode ser colorido com três cores. Verifique que vértices adjacentes possuem cores distintas.

a) Descreva uma heurística gulosa para construir uma solução para o PCG. Em sua resposta, deve haver um pseudocódigo e uma descrição textual do algoritmo.

b) Faça uma análise de complexidade de tempo do algoritmo proposto, fornecendo um limite O para o seu pior caso.

c) Discorra sobre os impactos das duas formas de representação de grafos estudadas na disciplina (matriz e listas de adjacência) na complexidade do algoritmo.