Teoria dos Grafos – COS242
2012/2

Sexta Lista de Exercícios

ATENÇÃO! Para ajudar no treinamento para as provas faça as listas de forma que todas as respostas estejam devidamente comentadas.

Questão 1: Considere o problema de escalonamento de tarefas em um intervalo (*interval scheduling*) apresentado em aula. Dê um contra-exemplo para mostrar que algoritmos gulosos baseados nas idéias abaixo não produzem a solução ótima para o problema em todos os casos.

1. Escalonar primeiro a tarefa que começa primeiro.
2. Escalonar primeiro a menor tarefa.
3. Escalonar primeiro a tarefa com o menor número de conflitos.
4. Descartar primeiro a tarefa com o maior número de conflitos.

Questão 2: Considere o problema de escalonamento de tarefas em um intervalo (*interval scheduling*) apresentado em aula. Descreva em pseudo-código o algoritmo guloso que retorna uma solução ótima para este problema. Analise a complexidade do tempo de execução do seu algoritmo.

Questão 3: Considere o problema de coloração de vértices de um grafo $G = (V, E)$ de forma que vértices adjacentes não podem ter a mesma cor.

1. Escreva em pseudo-código um algoritmo guloso para colorir os vértices (incluindo o passo de determinar a cor para um vértice). Determine a complexidade do tempo de execução do seu algoritmo.
2. Dê um contra-exemplo mostrando que o algoritmo guloso que colori os vértices na ordem decrescente de grau não produz a coloração ótima, ou seja, não utiliza o menor número de cores possível.
3. Qual é o menor número de cores necessário para colorir o mapa da América do Sul (ver grafo nos slides da aula).
4. Dê um exemplo de um mapa de regiões onde são necessárias quatro cores para colori-lo.

Questão 4: Considere o problema de clusterização de objetos.

1. Defina explicitamente uma função de similaridade (ou seja, distância) para imagens. Assuma que a imagem o_k, para $k = 1, \ldots, n$, onde n é o número de imagens, tem dimensões x_k por y_k, e que a cor do pixel na coordenada (i, j) é dada por $c_k(i, j)$ para $i = 1, \ldots, x_k$ e $j = 1, \ldots, y_k$. Sua função de similaridade deve utilizar os valores dos pixels das imagens assim como as dimensões das imagens.

2. Escreva em pseudo-código uma função para calcular o espaçamento de uma dada clusterização. Lembrando que o espaçamento de uma dada clusterização é a menor distância entre dois objetos de clusters diferentes. A entrada para seu algoritmo é a alocação dos n objetos em um conjunto de clusters $C_1, \ldots, C_m$, onde m é o número de clusters. Determine a complexidade do tempo de execução do seu algoritmo.

3. Escreva em pseudo-código uma função para calcular a maior distância entre dois objetos de um mesmo cluster. A entrada para seu algoritmo é a alocação dos n objetos para um conjunto de clusters $C_1, \ldots, C_m$, onde m é o número de clusters. Determine a complexidade do tempo de execução do seu algoritmo.

4. Dê um contra-exemplo onde a clusterização para $m = 2$ obtida via MST é ótima para o caso da métrica do item 2 e não é ótima para métrica do item 3. Ou seja, a clusterização do seu exemplo maximiza o espaçamento, mas não minimiza a maior distância entre dois objetos de um mesmo cluster. Seu contra-exemplo deve conter os objetos e as distâncias entre eles.