Construção e Análise de Algoritmos

lista de exercícios 20 - solução

Nota: Nem todos os exercícios dessa lista envolvem estratégias de aproximação (i.e., em alguns casos você poderá conseguir soluções exatas).

Além disso, em vários casos, não é preciso muita inteligência para alcançar o fator de aproximação indicado — mas, você pode tentar usar a sua inteligência para obter um fator de aproximação melhor do que o indicado.

1. **Potências de 2**

   O problema da partição de conjuntos fica mais fácil quando nós assumimos que todos os números são potências de 2?

   a) Apresente um algoritmo para essa versão do problema, assumindo que os números são todos distintos.

   b) Apresente um algoritmo para essa versão do problema, assumindo que podem haver números repetidos. Mais especificamente, assuma que pode haver uma quantidade arbitrária de cópias de cada número.

   c) Analise a qualidade das soluções produzidas pelos algoritmos dos ítens (a) e (b).

2. **Arranjo linear**

   Considere um grafo direcionado G com n vértices.

   O problema consiste em arranjar os vértices do grafo em uma sequência linear

$$v_1,\ v_2,\ v_3,\ \ldots,\ v_{n-1},\ ,v_n$$

   de modo que o número de arestas da forma

$$v_i \longrightarrow v_{i+k}$$

   (i.e., arestas apontando para a direita) seja o maior possível.

   Apresente um algoritmo eficiente para o problema com fator de aproximação 2.

   Isto é, você deve argumentar que o número de arestas apontando para a direita na solução ótima é no máximo 2 vezes maior do que o número de arestas apontando para a direita na sua solução.

solução:

Denote o número de arestas no grafo por m.

Claramente, $OPT \leq m$.

Agora, note que, para qualquer solução com valor k, se nós invertemos a ordem dos vértices nessa solução, nós obtemos uma solução com valor $m - k$.

Ora, certamente k ou $m - k$ é maior ou igual a $m/2$.

Portanto, a solução associada ao maior deles tem tamanho ao menos a metade da solução ótima.

$\diamond$

3. **A festa**

Alice quer dar uma festa e precisa decidir quem ela vai convidar.

Ela faz uma lista das n pessoas que ela conheça, e uma outra lista com os pares de pessoas que se conhecem.

Ela imagina que a festa seria mais divertida se todos os convidados:

- conhecessem ao menos outros 5 convidados da festa
- não conhecessem ao menos outros 5 convidados da festa

Por outro lado, ela gostaria de convidar o maior número possível de pessoas para a festa.

a) Apresente um algoritmo eficiente para esse problema.

b) Argumente que o seu algoritmo encontra uma solução ótima para o problema.

c) Estime a complexidade do seu algoritmo em termos dos comprimentos n, m das listas de Alice.

solução:

Etapa 1:

- *percorrer a lista de pares e*
  - *calcular o número de conhecidos de cada pessoa*
  - *construir uma lista de conhecidos para cada pessoa*

Etapa 2: (laço)

- *eliminar do problema aqueles que não satisfazem as restrições*
- *atualizar o número de conhecidos de cada pessoa*

Análise:

- *A etapa 1 claramente executa em tempo $O(m)$, onde m é o tamanho da lista de pares de conhecidos.*
- *A etapa 2 também executa em tempo $O(m)$.*

  Para ver isso, note que para cada pessoa eliminada nós devemos percorrer a sua lista de conhecidos

$\diamond$

4. **Partição de grafos**

Considere um grafo G com pesos associados às arestas.

Suponha que nós queremos particionar os vértices de G em dois subconjuntos A e B de modo que a soma dos pesos das arestas que cruzam a partição seja a maior possível.

Mais precisamente, defina

$$P(A,B) \quad = \sum_{\substack{(u,v) \in G \\ u \in A, v \in B}} \mathsf{peso}(u,v)$$

O objetivo do problema consiste em encontrar uma partição A, B cujo valor $P(A,B)$ é o maior possível.

a) Apresente um algoritmo para esse problema com fator de aproximação $1/2$.

   Isto é, você deve argumentar que a solução S encontrada pelo seu algoritmo satisfaz a seguinte condição

$$P(S) \quad \geq \quad \frac{1}{2} \cdot P(S_{ot})$$

   onde S_{ot} é uma solução ótima para o problema.

b) Generalize o seu algoritmo para o caso em que os vértices do grafo são particionados em 3 subconjuntos A, B, C.

   Qual o fator de aproximação do seu algoritmo nesse caso?

solução:

Escolha uma aresta (u,v) qualquer do grafo, coloque u em A e v em B.

A seguir, para cada outro vértice w

- *calcule*

$$P(w,A) \;=\; \sum_{(w,u), u \in A} \mathsf{peso}(w,u) \qquad\qquad P(w,B) \;=\; \sum_{(w,v), v \in B} \mathsf{peso}(w,v)$$

- *se* $P(w, A) > P(w, B)$ *coloque* w *em* B, *senão coloque* w *em* A

Argumento:

Seja P *o peso total das arestas do grafo.*

Seja $u, v, w_1, \ldots, w_n$ *a ordem em que os vértices foram considerados pelo algoritmo.*

E seja P_{w_i} *a soma total dos pesos das arestas que ligam* w_i *a algum vértice que aparece antes dele na sequência acima.*

Claramente,

- $P \quad = \quad \mathsf{peso}(u, v) \; + \; \sum_i P_{w_i}$
- $P(S) \quad \geq \quad \mathsf{peso}(u, v) \; + \; \sum_i \frac{P_{w_i}}{2}$

$\diamond$