Algoritmos Probabilísticos

Segundo semestre de 2006

Lista 1

1. Considere o algoritmo de busca aleatória visto na primeira aula.
   (a) Calcule o número esperado de iterações desse algoritmo caso o elemento x não apareça nenhuma vez no vetor.
   (b) Calcule o número esperado de iterações desse algoritmo caso o elemento x apareça exatamente k vezes no vetor.

2. Considere a variante do algoritmo de busca aleatória que, a cada iteração em que x não é encontrado, remove a posição visitada do vetor. Calcule o número esperado de iterações desse algoritmo caso o elemento x apareça exatamente uma vez no vetor. Calcule o número esperado de iterações desse algoritmo caso o elemento x não apareça nenhuma vez no vetor.

3. Mostre que $e^x \geq 1 + x$ para todo x, com igualdade se e somente se $x = 0$.

4. Descreva uma implementação $O(n^2)$ (de uma execução) do algoritmo para calcular um corte mínimo visto em aula. Aqui n denota o número de vértices do grafo dado. Diga exatamente quais EDs você usaria, como implementaria a contração de uma aresta e como faria o sorteio de uma aresta sobrevivente com probabilidade uniforme, etc. Analise a complexidade de tempo de cada etapa. Conclua que, se executarmos o algoritmo $O(n^2 \lg n)$ vezes, obtemos um algoritmo $O(n^4 \lg n)$ que devolve um corte mínimo em um dado grafo com n vértices com probabilidade pelo menos $1 - 1/n$.

5. Considere a variante do problema do corte mínimo visto em aula que tem como entrada um grafo G com um custo c_e inteiro não-negativo para cada aresta e do grafo. Adapte o algoritmo visto em aula para essa variante do problema. Calcule (uma delimitação para) a probabilidade de uma execução do seu algoritmo devolver um corte de custo mínimo em G. Calcule o número de vezes que você precisaria executar esse algoritmo para obter um curto de custo mínimo com probabilidade maior que $1 - 1/n$, onde n é o número de vértices de G.

6. Considere a variante do algoritmo visto em aula para encontrar um corte mínimo, que prossegue contraindo arestas até que o grafo remanescente tenha t vértices, em vez de dois. Neste ponto, a variante devolve o grafo remanescente. Calcule a probabilidade de que um particular corte mínimo C sobreviva a estas contrações, ou seja, calcule a probabilidade de que C seja ainda um corte no grafo remanescente.

7. Considere a variante do algoritmo para encontrar um corte mínimo visto em aula que consiste no seguinte. Dado um grafo G, se G tem no máximo 6 vértices, encontre um corte mínimo em G por exemplo na força bruta e o devolva. Senão, execute a variante descrita no exercício anterior para $t = \lceil 1 + n/\sqrt{2} \rceil$ duas vezes independentemente obtendo dois grafos, G_1 e G_2. Aplique o algoritmo recursivamente a G_1 e G_2, obtendo dois cortes C_1 e C_2, respectivamente. Devolva o menor entre C_1 e C_2.

   (a) Qual é a profundidade da recursão desse algoritmo?

   (b) Mostre que as operações do algoritmo, excluídas as chamadas recursivas, consomem tempo $O(n^2)$. Deduza a complexidade do algoritmo.

   (c) Denote por $P(n)$ a probabilidade desse algoritmo, quando executado com um grafo de n vértices como entrada, devolver um corte mínimo do grafo. Argumente que $P(n)$ satisfaz a seguinte recorrência:

   $$P(n) \;\geq\; 1 - (1 - \frac{1}{2}P(\lceil 1 + n/\sqrt{2} \rceil))^2.$$

   (d) Resolva a recorrência acima. Você consegue? Deixa eu dar uma ajudinha... Primeiro faça uma mudança de variável e considere P em função da profundidade da árvore de recursão em vez de em função do número de vértices do grafo. Ou seja, denote por k a profundidade da árvore de recursão gerada quando da chamada do algoritmo para um grafo G. Seja p a função resultante. Então a recorrência para p em função de k fica:

   $$p(k) \;\geq\; 1 - (1 - \frac{1}{2}p(k-1))^2,$$

   sendo $p(0) = 1$, já que, na base, o algoritmo sempre devolve um corte mínimo. Observe que a fórmula acima pode ser reescrita resultando em

   $$p(k) \;\geq\; p(k-1) - \frac{p(k-1)^2}{4}.$$

   Faça mais uma troca de variáveis, reescrevendo a recorrência acima em função de $q(k)$, onde $q(k) = 4/p(k) - 1$, ou seja, $p(k) = 4/(q(k) + 1)$. Mostre por indução que $k < q(k) < k + H_{k-1} + 3$. Deduza o comportamento assintótico de $p(k)$ e depois de $P(n)$.

Notas: O algoritmo simples visto em aula adaptado para o caso com custos foi proposto por Karger [SODA 1993], enquanto que a variante mais rápida descrita acima foi proposta por Karger e Stein [STOC 1993].