Algoritmos Probabilísticos

Segundo semestre de 2006

Lista 4

1. Uma moeda honesta é jogada n vezes. Para $1 \leq i < j \leq n$, seja $X_{ij} = 1$ se o i-ésimo e o j-ésimo lançamento da moeda resultaram na mesma saída, e $X_{ij} = 0$ caso contrário. Mostre que as variáveis X_{ij} são 2-a-2 independentes, mas que não são totalmente independentes.

2. (a) Sejam X e Y dois números que são escolhidos independente e uniformemente do conjunto $\{0, 1, \ldots, n\}$. Seja Z a soma de X e Y módulo $n+1$. Mostre que X, Y e Z são 2-a-2 independentes, porém não totalmente independentes.

   (b) Estenda esse exemplo para dar uma coleção de variáveis que são k-a-k independentes porém não $(k+1)$-a-$(k+1)$ independentes.

3. Prove que a definição de BPP com $1 - 2^{-n^c}$ e 2^{-n^c} no lugar de $3/4$ e $1/4$ respectivamente é equivalente à original, onde c é uma constante qualquer.

4. Mostre que a seguinte definição alternativa é equivalente à definição de FPRAS dada numa das aulas: um FPRAS para um problema é um algoritmo probabilístico para o qual, dada uma entrada x e qualquer parâmetro ϵ com $0 < \epsilon < 1$, o algoritmo produz uma $(\epsilon, 1/4)$-aproximação em tempo polinomial em $1/\epsilon$, $\lg 1/\delta$ e no tamanho da entrada x. (*Dica:* amplie a probabilidade de sucesso de $3/4$ para $1-\delta$, considerando a *mediana* de diversas execuções independentes do algoritmo. Por que a mediana é melhor que a média?)

5. Suponha que temos uma classe de instâncias do problema DNF, cada uma com $\alpha(n)$ atribuições que a satisfazem, onde α é uma função polinomial e n é o número de variáveis da fórmula. Suponha que aplicamos o algoritmo ingênuo de amostragem de atribuições e verificamos se elas satisfazem a fórmula dada. Mostre que, amostrando $2^{n/2}$ atribuições, a probabilidade de encontrar pelo menos uma atribuição que satisfaça a dada fórmula é exponencialmente pequena em n.

6. (a) Sejam $S_1, \ldots, S_m$ subconjuntos de um conjunto U. Conhecemos $|S_i|$ para cada $1 \leq i \leq m$. Queremos uma (ϵ, δ)-aproximação para o tamanho do conjunto

   $$S = \cup_{i=1}^{m} S_i.$$

   Temos uma rotina disponível que pode, em um passo, escolher um elemento aleatória e uniformemente de um conjunto S_i. Também, dado um elemento $x \in U$, podemos determinar o número de conjuntos S_i para os quais $x \in S_i$. Chamamos esse número de $c(x)$.

   Seja $p_i = |S_i| / \sum_{j=1}^{m} |S_j|$. A j-ésima amostra consiste dos seguintes passos. Escolha um conjunto S_j, com a probabilidade de cada conjunto S_i sendo p_i, e então escolha um elemento x_j aleatória e uniformemente de S_j. Em cada amostra, as escolhas aleatórias são independentes das escolhas das demais amostras. Depois de t amostras, estimamos $|S|$ por

$$\left(\frac{1}{t}\sum_{j=1}^{t}\frac{1}{c(x_j)}\right)\left(\sum_{i=1}^{m}|S_i|\right).$$

Determine — como uma função de m, ϵ e δ — o número de amostras necessárias para obter uma (ϵ, δ)-aproximação para $|S|$.

(b) Explique como usar os resultados da parte (a) para obter um algoritmo de aproximação para calcular o número de soluções de uma fórmula DNF.

7. Considere o algoritmo BIT-FIXING (o que usa a rotina BIT-FIXING) para roteamento de uma permutação no hipercubo de dimensão n. Suponha que n é par. Escreva cada vértice fonte s como a concatenação de duas cadeias binárias a_s e b_s, cada uma de comprimento $n/2$. Tome como vértice destino de s a concatenação de b_s e a_s. Mostre que essa permutação faz o algoritmo BIT-FIXING gastar $\Omega(\sqrt{N})$ unidades de tempo.

8. Considere a seguinte modificação do algoritmo BIT-FIXING para rotear uma permutação no hipercubo de dimensão n. Suponha que, em vez de corrigir os bits na ordem de 1 a n, cada pacote escolhe uma ordem aleatória (independente da escolha dos demais pacotes) e corrige os bits nesta ordem. Mostre que existe uma permutação para a qual este algoritmo exige $2^{\Omega(n)}$ passos com alta probabilidade.

9. Suponha que usamos o algoritmo probabilístico de duas fases para rotear no hipercubo um total de $p2^n$ pacotes, onde cada vértice é a fonte de no máximo p pacotes e o destino de no máximo p pacotes.

   (a) Dê uma delimitação que vale com alta probabilidade para o tempo de pior caso do algoritmo.

   (b) Dê uma delimitação que vale com alta probabilidade para o maior número de pacotes em qualquer vértice a qualquer momento da execução do algoritmo.

10. Considere a seguinte variante do algoritmo probabilístico de duas fases para roteamento no hipercubo. Um inteiro a é escolhido aleatória e uniformemente do conjunto $\{0, \ldots, N-1\}$ e então o vértice intermediário associado ao vértice i fica sendo o vértice $(a+i) \bmod N$ para todo i. (Apenas um a é escolhido, e não um a para cada i.) Prove que, para cada aresta e do hipercubo, nas duas fases, o número esperado de pacotes que atravessa a aresta e é constante.