PROVA 1 DE ESTRUTURAS DE DADOS
BCC, 1o. SEMESTRE DE 2013

Instruções:

1. Não destaque as folhas do caderno de soluções.
2. A prova pode ser feita a lápis. Cuidado com a legibilidade.
3. Não é permitido o uso de folhas avulsas para rascunho.
4. Não é necessário apagar rascunhos no caderno de soluções.
5. Asserções imprecisas valem pouco. Justifique suas asserções (dentro do razoável!).
6. **Duração da prova: 1 hora e 40 minutos**

1. [5 pontos] Esta questão é sobre seu EP1. Particularmente importante nesta questão é que você escreva de forma sucinta, clara, e informativa. Você não perderá nota por algoritmos ou estruturas de dados não muito eficientes.
   (i) Dê uma descrição de *alto nível* das estruturas de dados e algoritmos que você implementou para resolver o EP1. Naturalmente, diga como estas estruturas e algoritmos foram usados. Lembre que seu EP1 tinha dois modos de operação (o modo de verificação de conexidade, com o valor de d dado, e o modo de determinação do menor d que torna um certo grafo conexo). Descreva suas estruturas de dados e algoritmos em cada caso separadamente.
   (ii) Faça uma discussão sobre a eficiência de sua solução do EP1. Quanto tempo seu programa demora para gerar N pontos ao acaso e decidir se um dado d define um grafo conexo?
   (iii) Quanto tempo seu programa demora para gerar N pontos ao acaso e encontrar o menor d que define um grafo conexo? Tanto em (ii) como em (iii), você pode dar sua resposta usando a notação O.
   (iv) Você conhece outras formas de se resolver o EP1? Como é a complexidade dessas outras soluções? (Faça uma análise como em (ii) e (iii).)
2. [5 pontos] Sejam dados inteiros M e N, com $0 \leq N \leq M$. Queremos gerar N inteiros $x_1, \ldots, x_N$ em $S = \{0, 1, \ldots, M-1\}$, *sem repetição.* Ademais, queremos que o conjunto $\{x_1, \ldots, x_N\}$ assim gerado esteja uniformemente distribuído entre todas as $\binom{M}{N}$ possibilidades (isto é, cada conjunto de N elementos contido em S deve ter probabilidade $\binom{M}{N}^{-1}$ de ser gerado).
   (i) Dê uma descrição de alto nível (mas precisa) de um algoritmo eficiente para resolver este problema, no caso em que $M \leq 10^5$ e $N \leq M/3$.
   (ii) Dê uma descrição de alto nível (mas precisa) de um algoritmo eficiente para resolver este problema, no caso em que M pode ser bastante grande, como por exemplo $M = 2^{31} - 1$, mas ainda temos $N \leq M/3$. [*Sugestão.* Use que podemos implementar

---

Date: Versão de 24 de junho de 2013.

tabelas de símbolos de forma que as operações de inserção e busca consomem, 'tipicamente', tempo $O(\log N)$ quando a tabela contém N elementos. Diga como são feitas tais implementações.]

(iii) Seu algoritmo em (i) roda em tempo $O(M + N)$? Seu algoritmo em (ii) roda em tempo $O(N \log N)$? Justifique suas respostas. [*Observação.* Você pode supor conhecido que se $N \leq M/3$ e uma sequência de $3N$ inteiros de S são gerados uniformemente ao caso, então, com alta probabilidade, pelo menos N inteiros distintos ocorrem na sequência.]

(iv) Quanta memória gastam seus algoritmos em (i) e (ii)? Responda para N e M genéricos (em particular, não leve em conta a restrição $M \leq 10^5$ em (i)). Você pode responder usando a notação O. Justifique.

(v) Responda (iv) para os casos concretos $(N, M) = (10^4, 10^6)$, $(N, M) = (10^5, 10^6)$ e $(N, M) = (10^5, 10^9)$ (sua resposta pode ser (bastante) aproximada e deve ser dada em bytes, kilobytes, megabytes, etc). Justifique.

(vi) Diga como resolver (i) e (ii) removendo a restrição $N \leq M/3$. [*Observação.* Você pode supor conhecido que se uma sequência de M inteiros de S são gerados uniformemente ao caso, então, com alta probabilidade, em torno de $M/e = M/2.71828\ldots$ inteiros distintos ocorrem na sequência (lembre-se dos exemplos disponibilizados na página da disciplina).]

Nesta questão, você deve supor que você tem acesso a um gerador de números aleatórios `unif_rand()` que devolve um inteiro em S, com todos esses inteiros equiprováveis. Suponha que cada chamada de `unif_rand()` consome tempo constante.