Tópicos Avançados em Algoritmos - prova II – inclui uma resolução

1. Diga se cada uma das seguintes afirmações é verdadeira ou falsa (não é necessário justificar)

a. Seja qual for o vector de elementos a ordenar, o "quick-sort" aleatorizado tem um tempo médio de execução de ordem $O(n \log n)$. v

b. Existe um algoritmo *determinístico* polinomial para determinar se um inteiro n é primo. v

c. Para qualquer i, o elemento de ordem i de um vector (não necessariamente ordenado) pode ser determinado em tempo $O(\log n)$ f

d. A mediana de um vector já ordenado pode ser determinada em tempo $O(1)$. v

e. A profundidade de uma rede de comparadores com 4 entradas ($n = m = 4$) e 3 comparadores não pode exceder 2. f

f. A profundidade (tempo paralelo) de uma rede de ordenação de n entradas pode ser de ordem $\Omega(n^3)$. v

2. Defina a classe de linguagens RP. Defina a classe de linguagens BPP. Mostre que, se uma linguagem pertence à classe RP, também pertence à classe BPP.

→Uma linguagem L pertence à classe RP se existir um polinómio $l(n)$ uma função $f(x, y)$ computável em tempo polinomial em $|x|$ tal que

$$\begin{cases} x \in L & \Rightarrow & \text{prob}_{\ y \in_u \{0,1\}^{l(n)}}[f(x,y) = 1] \geq 1/2 \\ x \notin L & \Rightarrow & \text{prob}_{\ y \in_u \{0,1\}^{l(n)}}[f(x,y) = 1] = 0 \end{cases}$$

e uma linguagem L pertence à classe BPP se existir um polinómio $l(n)$ e uma função $f(x, y)$ computável em tempo polinomial em $|x|$ tal que

$$\begin{cases} x \in L & \Rightarrow & \text{prob}_{\ y \in_u \{0,1\}^{l(n)}}[f(x,y) = 1] \geq 3/4 \\ x \notin L & \Rightarrow & \text{prob}_{\ y \in_u \{0,1\}^{l(n)}}[f(x,y) = 1] \leq 1/4 \end{cases}$$

(i) Suponhamos que L está na classe RP e seja $f(x, y)$ a função que lhe está associada. Como foi descrito num exercício e nos apontamentos, a função $f_1(x, y)$ assim definida[1]

```
f1(x,y):
    y <- aleatório uniforme de n bits
    if f(x,y): return 1
    y <- aleatório uniforme de n bits
    if f(x,y): return 1
    return 0
```

que satisfaz

$$\begin{cases} x \in L & \Rightarrow & \text{prob}_{\ y \in_u \{0,1\}^{l(n)}}[f_1(x,y) = 1] \geq 3/4 \\ x \notin L & \Rightarrow & \text{prob}_{\ y \in_u \{0,1\}^{l(n)}}[f_1(x,y) = 1] = 0 \end{cases}$$

Como $0 < 1/4$, a existência da função f_1 demonstra que L também pertence à classe BPP.

3. Pretende-se ordenar um vector com n inteiros, eventualmente repetidos, compreendidos entre 1 e 1000. Descreva numa linguagem de programação informal um algoritmo com eficiência $O(n + u)$ para esse fim. →Algoritmo:

```
    DADO: v, RESULTADO:w
    v[1..n]: vector a ordenar
    w[1..n]: vector ordenado
    c[1..u], c[i]: número de elementos em v[] iguais a i (u=1000)

1   for i=1 to u: c[i]=0        //  O(u)
2   for i=1 to n:               //  n+1 testes
3     c[v[i]] = c[v[i]] + 1     //  n atribuições
4   for i=1 to u:               //  u+1 testes
5     for j=1 to c[j]:          //  n+u testes
6       w[k]=j                  //  n atribuições
7       k = k+1                 //  n atribuições
    return w                    //  vector ordenado
```

Somando as contribuições das 7 linhas, vê-se que a complexidade é $O(n+u)$. Exercício. Verifique que o número de testes na linha 5 é $n + u$.

[1] Usando constantes apropriadas (por exemplo, 3/4 na definição de RP, em vez de 1/2) não teríamos necessidade de executar f mais que uma vez.

4. Defina o problema da selecção. Descreva em pseudo-código uma função `sel(v[a..b],i)` (a sua chamada é `sel(v[1..n],i)`) correspondente ao problema da selecção; essa função deve ser "inspirada" no "quick-sort" e ter tempo médio de ordem $O(n)$. Supondo que n é da forma $2^p - 1$ e que as divisões do intervalo `[1..n]` são em partes iguais[2], mostre que o número de comparações efectuadas é de ordem $O(n)$.

→Definição de "selecção" e algoritmo: ver apontamentos.
Eficiência do algoritmo na hipótese enunciada: $n = 2^p - 1$, divisões em partes iguais:
Se o intervalo tem m elementos ($m = n$ no início): (i) o "split" efectua $m-1 < m$ comparações, (ii) cada sub-intervalo tem comprimento inferior a $m/2$ (o valor exacto é $(m-1)/2$). Usando estes 2 factos, vê-se que o número total de comparações satisfaz

$$c(n) \leq n + n/2 + n/4 + \ldots + 1 < n + n/2 + n/4 + \ldots = 2n \in O(n)$$

5. Considere um circuito com n entradas e m saídas. Defina "depth" (profundidade ou tempo paralelo) do circuito. Suponha que os circuitos elementares usados no circuito têm exactamente 2 entradas e que todas as saídas dependem das n entradas[3]. Mostre que a profundidade do circuito é pelo menos $\lceil \log n \rceil$.
[Nota informativa. São exemplos de circuitos deste tipo: as redes de ordenação,
$y = \mathrm{xor}(x_1, \ldots, x_n)$, $y = x_1 \wedge x_2 \wedge \ldots \wedge x_n$.]

→Ver a definição de "depth" nos apontamentos.
Vamos começar por mostrar que, se o nó y tem profundidade d, então ele depende, no máximo, de 2^d entradas. O caso base $d = 0$ é trivial: nesse caso y é uma entrada e depende de $2^0 = 1$ entradas. Suponhamos agora que $d \geq 1$, sendo $y = f(w, z)$ onde f é um circuito elementar. Por definição de profundidade, cada uma das 2 entradas (w e z) de f tem profundidade no máximo $d - 1$ e portanto, pela hipótese indutiva, cada uma das 2 entradas de y depende no máximo de 2^{d-1} entradas do circuito; assim, y depende no máximo de $2^{d-1} + 2^{d-1} = 2^d$ entradas do circuito. □
Se o número de entradas do circuito é n e a profundidade de y é d, temos então $n \leq 2^d$, ou seja, $d \geq \log n$. Como d é necessariamente inteiro, vem $d \geq \lceil \log n \rceil$.

[2] Exemplo de divisões em partes iguais para $n = 15 = 2^4 - 1$: tamanho dos sucessivos intervalos $15 \to 7 \to 3 \to 1$.

[3] No sentido seguinte: seja $y(x_1, \ldots, x_n)$ uma saída; para qualquer i com $1 \leq i \leq n$, existem valores $x_1, \ldots x_{i-1}$, $x_{i+1}, \ldots, x_n$, a e b tais que $y(x_1, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_n) \neq y(x_1, \ldots, x_{i-1}, b, x_{i+1}, \ldots, x_n)$. Em termos intuitivos, a entrada de ordem i pode infuenciar qualquer saída y.