

Cartão: 253124

Nome: Eliza Emanuel Bernart

1. (2.5 pontos) O que é um *modelo de computação*? Para que serve um modelo de computação? Dê 2 exemplos de modelos de computação, explicando cada um deles e deixando evidente como "*computação*" é definida em cada modelo. Qual o significado de dizer "*Os modelos de computação A e B são equivalentes*"? Os modelos que você citou são equivalentes? Justifique (não precisa fazer uma prova formal, somente explicar o que deve ser feito para provar a equivalência/não-equivalência entre estes modelos).
2. (2.5 pontos) Um desenvolvedor de software precisa construir um programa para solucionar um problema X. Procurando em uma biblioteca, ele encontra um programa eficiente (com ordem de complexidade $O(n^2)$) que soluciona um problema PE que parece ser muito similar ao problema X. Por outro lado, ele encontra um texto que fala sobre um problema PNC, dizendo que ele não é computável, e o desenvolvedor acha que PNC também parece similar a X.
 - (a) Explique o que é uma redução e como definimos reduções entre problemas. Quando dizemos que uma redução é polinomial?
 - (b) Diga se as afirmações a seguir são verdadeiras ou falsas e justifique:
 - (A) O problema PE não é NP-completo.
 - (B) Se conseguimos reduzir X para PNC, X também não é computável.
 - (C) Se conseguimos reduzir X para PE, X pertence à classe P.
3. (1.5 pts): É fornecido um array T com $n + 2$ entradas que representam valores de temperaturas. As posições 0 e $n + 1$ de T são inicializadas com $-\infty$. A temperatura é medida de 30 em 30 min, e cada vez que qualquer mudança de temperatura é detectada nas medições, a posição seguinte de T é preenchida com a nova temperatura. Suponha que ao longo do dia as temperaturas só aumentam até um certo pico, e então posteriormente só decrescem. Dado T com n entradas preenchidas, projete um algoritmo que retorne a maior temperatura medida do dia, acessando o menor número possível de entradas de T . Analise a complexidade de pior caso do seu algoritmo.

Por exemplo, para o vetor abaixo, o algoritmo retornaria o valor 10.1.

$$T = [-\infty \quad 9.7 \quad 9.9 \quad 10.1 \quad 10.0 \quad 9.7 \quad 9.5 \quad 8.9 \quad 8.7 \quad 8.6 \quad 8.2 \quad 8.1 \quad -\infty]$$
4. (2 pts) Uma empresa de telecomunicações decidiu minimizar seus gastos. Sua infraestrutura pode ser representada por um grafo $G = (V, A)$ não direcionado e com um valor positivo atribuído a cada link representando o custo (manutenção e aluguel) daquele link. Dada a infraestrutura atual, a empresa deseja desativar o maior número possível de links de forma que a rede resultante $G' = (V, A')$ possua o mesmo conjunto de vértices original V , mas com um conjunto $A' \subseteq A$ de forma que a rede ainda permita um caminho viável entre cada par de vértices. Em outras palavras, será mantida uma rede conexa de menor custo. Dado o grafo G como entrada, projete um algoritmo que retorne o grafo G' . Analise a complexidade de pior caso do seu algoritmo.

5. (1.5 pts) Analise a complexidade de pior caso e de melhor caso do algoritmo quicksort. A entrada dada por A, p, r é um vetor de números A , e índices p e r . Na primeira chamada do algoritmo $p=1$ e $r=n$. Informe detalhes da sua análise na resolução da questão.

```
1: function QUICKSORT(A,p,r)
2:   if p < r then
3:     q = partition(A,p,r);
4:     Quicksort(A,p,q-1);
5:     Quicksort(A, q+1,r);
6:   end if
7: end function
```

```
1: function PARTITION(A,p,r)
2:   x=A[r];
3:   i=p-1;
4:   for j=p to r-1 do
5:     if A[j] ≤ x then
6:       i = i+1
7:       trocar A[i] e A[j];
8:     end if
9:   end for
10:  trocar A[i+1] e A[r];
11:  return(i+1);
12: end function
```
