complexidade $O(n \lg n)$

correção armazenar todos os pontos dentro de um intervalo unitário que vai sobre a partir do 1º ponto

PROVA 1999/2

Questão 1 - a) O algoritmo em seu pior caso pode a partir de uma entrada de tamanho n consumir tempo $n^2 \lg n$.

b) O algoritmo a partir de uma entrada de tamanho n consome no mínimo n^2 unidades de tempo.

Questão 2 - Deve-se dividir o vetor sempre na metade para que a pilha de recursão tenha tamanho $O(\log n)$

Sendo $k = \log n + 1$

$n = r - p + 1$

$\frac{n}{2} = r_1 - p_1 + 1$

$\frac{n}{2^2} = r_2 - p_2 + 1$

$\frac{n}{2^k} = r_k - p_k + 1$

} $\log n$

Questão 3 - a) heap é uma árvore binária onde o nó pai i tem prioridade em relação aos filhos $2i$ e $2i+1$. O heap pode ser um MIN-HEAP onde a chave $[\lfloor i/2 \rfloor] \leq$ chave $[i]$ ou um MAX-HEAP onde chave $[\lfloor i/2 \rfloor] \geq$ chave $[i]$.

A construção de um heap é feito em tempo linear

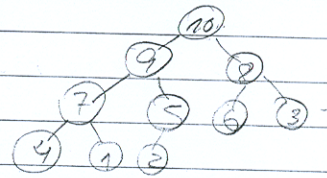
```
Build Heap (A, n)
  Para i ← ⌊n/2⌋ decrescendo até 1
      MAX/MIN HEAPIFY (A, i)
```

A rotina MAX/MIN Heapify ajusta os nós de forma que o pai tenha prioridade sobre os nós filhos.

b) 10 5 8 4 9 6 3 7 1 2

9 7 5 4

10 9 8 7 5 6 3 4 1 2



Questão 4 - Estrutura de dados: lista encadeada

a) Um nó nulo pode ser criado em tempo constante

b) A inclusão de um nó pode ser feita em tempo constante apenas alterando ponteiros dos nós vizinhos.

c) A exclusão de um nó tb pode ser feita em tempo constante já que é dado sua localização. Tendo apenas que alterar ponteiros nos vizinhos.

Questão 5 -

|1|0|0|0| |0|1|0|0| → |0|0|1|0| $n=4$

0 1 1 1 0 0 1

0 0 1 1 1 0 1

$j \leftarrow 1$

Para $i \leftarrow 2$ até n

  se $A[i] < A[j]$

    $A[j] \leftrightarrow A[i]$

    $j \leftarrow j+1$ } $n-1$

  Senão se $A[i] > A[j]$

    $j \leftarrow j+1$

Questão 6 - $n=4$

$Ast(3) + 4 + Ast(3)$

$Ast(2) + 3 + 4 + Ast(2) + 3$

$Ast(1) + 2 + 3 + 4 + Ast(1) + 2 + 3$

$1 + 2 + 3 + 4 + 1 + 2 + 3$ (16)

$T(n) = 0$ se $n = 0$

$T(n) = 2T(n-1) + n$ $n-i=0$, $n=i$

$= 2(2T(n-2)+n-1) + n$

$= 2^2 T(n-2) + 2n - 2 + n$

$= 2^2(2T(n-3) + n - 2) + 2n - 2 + n$

$= 2^3 T(n-3) + 2^2 n - 2^3 + 2n - 2 + n$

$= 2^i T(n-i) + \sum_{k=0}^{i-1} 2^k (n-k)$

$= 2^n T(0) + \sum_{k=0}^{i-1} 2^k n - \sum_{k=0}^{i-1} 2^k \cdot k$

$= 0 + n(2^n - 1) - \left[\frac{n 2^{n+1}}{2-1} + \frac{1-2^n}{(2-1)^2}\right]$

$= 2^n n - n - n 2^{n+1} - 1 + 2^n$

$= 2^n n + 2^n - (2n^{n+1} + n + 1)$

$\leq 2^n(n+1)$

base. $n=0$

$T(0) = 0 \leq 2^0(0+1) = \leq 1$

passo $n > 0$

$T(n) = 2T(n-1) + n$

$\overset{HI}{\leq} 2 \cdot (2^{n-1}(n-1+1)) + n$

$= 2^n (n) + n$

$= 2^n \cdot n + n$

$\leq 2^n \cdot n + 2^n$

$= 2^n (n+1)$

Questão 7 - MOCHILA (p, v, x, P) errado olhar transp. pdf

1 ordene por valor/peso $\frac{v_1}{p_1} > \dots \frac{v_n}{p_n}$

2 peso ← 0 $M \leftarrow \emptyset$ $i \leftarrow 1$

3 Para $i \leftarrow 1$ até n

4   se peso + $p_i \leq P$

5     $M \leftarrow M \cup x_i$

6   peso ← peso + p_i

Tamanho da instância: n

$T(n)$ = tempo de consumo no pior caso

Linha

1 $O(n \lg n)$

2 $O(1)$

3-6 $O(n)$

$T(n) = O(n \lg n)$

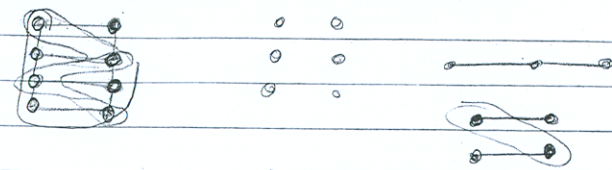
20   5 $\frac{5}{10} = \frac{1}{2}$ $10 \cdot \frac{1}{2} = 5$

Correção: Escolha gulosa:

teorema: Se para um elemento x_i $v_i/p_i = Max\{v_1/p_1 \dots v_n/p_n\}$ então ele está em uma solução ótima.

Questão 8 - emparelhamento

M de arestas cobertura



a) Vou provar que o problema a é verificável em tempo polinomial.

Prova: Seja C um conjunto de vértices de G, para cada aresta uv em G, o vértice u ou v pertence ao conjunto C. Isso pode ser feito em tempo polinomial.

b) Seja C um conjunto de vértices para cada aresta uv se u e v pertencem a C ou nem u nem v pertencem a C então C não é cobertura de G de Tamanho mínimo. Isso pode ser feito em tempo polinomial.

FIM PROVA 1999

ELRS 15.3-3 - Problema da Matriz - maximizar nº de multiplicações escalares. Possui propriedade de subestrutura ótima?

Sim. Se existe uma parentização ótima em $A_1 \dots A_n$ que maximize o nº de multiplicações então $A_1 \dots A_k$ e $A_{k+1} \dots A_n$ também são parentizações ótimas que maximizam nº de multiplicações.

Exe 10-1f - Discuta o seguinte caso especial do problema subset-sum. Dados inteiros não negativos a, b, m, n e W encontrar inteiros não-negativos $i \leq m$ e $j \leq n$ tais que $ia + jb = W$

Recorrência:

$C[k, v]$ { 1 se tem solução
0 caso contrário

$C[k,0] \leftarrow 1$

$C[0,v] \leftarrow 0$ se $v > 0$

$C[k,v] \leftarrow C[k-1,v]$ se $k.a > v$ ou $k.b > v$

$C[k,v] \leftarrow \max\{C[k-1,v], C[k-1, v-k.a], C[k-1, v-k.b]\}$