José Douglas Gondim Soares, 485347

2ª Avaliação Síncrona CANA

2-) a)

```
PE ← [1..M]
respostas ← []

1  Proc achaElementos(PE[b..m], V[i..j]){
2     Se (b < M) retorna
3     Se (b = M) {
4        resposta.addEmOrdem(selecao-DC(PE[K], V[i..j]))
5        retorna
6     }
7     K ← (M-b)/2
8     valor ← seleção-DC(PE[K], V[i..j])
9     resposta.addEmOrdem(valor)
10    P ← particao(valor, V[i..j])
11    achaElementos(PE[b, K], V[i..p])
12    achaElementos(PE[K+1, M], V[p+1..j])

}
```

Obs: A partição da linha 10 altera o vetor original para a versão particionada.

b) Análise de complexidade

No caso médio, em que todas as posições a serem procurados estão mais ou menos espaçadas, cada vez que eu seleciono um elemento intermediário do vetor PE, o tamanho da lista que eu vou precisar selecionar da próxima vez cai pela metade.

Continua..

sendo assim, em média eu particiono e seleciono a metade da lista original a cada chamada recursiva. logo, cada elemento procurado leva em média metade do tempo que o elemento procurado levou.

$= 2.(n + n/2 + n/4 ... + M) = 2. \log_2 n . M$

$= O(M . \log_2 n)$

C) Com a complexidade encontrada sim, pois mesmo com $n = M$, a complexidade ainda é igual a $n . \log_2 n$, que é a velocidade mais rápida que podemos ordenar um vetor com o mergesort, por exemplo. Isso não ocorre na versão ingênua $O(M . n)$, onde $M > \log_2 n$ já seria pior que ordenar o vetor inteiro.

3- Proc enc-menor(V[K..n]){
    menor ← K
    Para j ← K+1 até n
        Se (V[j] < V[menor])
            menor ← j
    retorna menor

I - Variáveis menor e j

II Para todo j, V[menor] <= V[j]

III - No início, menor = K e j não existe

IV - Se (V[j] < V[menor]) menor ← j

logo, para j = n V[menor] < V[j]

Procedimento inserção_Rec(V[K..n]){
    Se (K=n) retorna
    aux ← enc_menor(V[K..n])
    troca(aux, K)
    inserção_Rec(V[K+1..n])
}

I - Variáveis K, aux

II - K é o menor elemento do V[K...n]

III - No início se (K=n) ⇒ 1 elemento no vetor, retorna, pois K já é o menor. Considerando a corretude de enc_menor, aux será o menor e troca(aux, K), logo, K permanece o menor para K=1.

IV - Para K = n, K é incrementado em 1 a cada chamada recursiva, logo, K-1 permanece o menor elemento de V[K-1..n].

Eu, José Douglas Sandim Soares, declaro que não pesquisei na internet nem perguntei a colegas nada a respeito da prova. 31/08/20, prova de CANA.