7,8

Ana Keylla da Fonseca Sousa
SIII - Eng. Computação

Muito bom!

(01) Algoritmo computacional correto e eficiente o algoritmo que possui um número de instruções finita e não ambiguas, que executa corretamente todas as instâncias para qual foi desenvolvido. Sua efiência está em ele executar uma quantidade de instruções elementares onde o tempo é limitado ~~pelo polinômio~~ por um polinômio no tamanho ~~do polinômio~~ da entrada. 1,8

(02) B, E 1,0

(04) Algoritmo de busca

```
ENTRADA: LISTA DE n NÚMEROS E UM NÚMERO x
SAÍDA: SE O NÚMERO ESTÁ OU NÃO
PARA i=0 ATÉ n-1
    SE LISTA[i] = x
        DEVOLVA SIM
    FIMSE
DEVOLVA O
```

3,0

A sua região critica será (se lista[i] = x), analisando essa região podemos perceber que a complexidade temporal do algoritmo será $\Theta(n)$. Sua complexidade ~~temporal~~ espacial é $\Theta(1)$. Como sua complexidade temporal é limitada pelo tamanho do polinômio da entrada, logo ele é eficiente. Para provarmos que o algoritmo é correto por indução $x = a$ e teremos uma lista de n elementos, é facil ver que a lista será percorrida $n-1$ vezes indepedente do valor de n, devolvendo "sim" caso x estejá contido na lista e "O" caso não estejá contido na lista.

(03) Sua região critica está em (se a[i] = a[j]), analisando a sua região critica observamos que existem dois laços "para" de complexidade n, logo a complexidade temporal do algoritmo é $\Theta(n^2)$. Sua complexidade espacial será $\Theta(1)$. Como sua complexidade temporal é limitada pelo tamanho do polinômio de sua entrada, logo ele é eficiente. 2,0