Exercícios Teóricos – u01: Noções de Complexidade

Catarina F. M. Castro (803531) – AEDs II

# Exercícios Iniciais

Uma imagem contendo Texto

Descrição gerada automaticamente



1. Plotar os gráficos das respectivas funções

Gráfico

Descrição gerada automaticamente

1. Plotar os gráficos das respectivas funções

Gráfico

Descrição gerada automaticamente

# Contagem de Operações

## Estrutura sequencial e condicional

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

R: 3 subtrações

Texto

Descrição gerada automaticamente

R: 3 subtrações no melhor caso, e 5 no pior

Texto

Descrição gerada automaticamente

R: 5 subtrações no melhor caso e 7 subtrações no pior caso.

## Estrutura de repetição simples

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

R: 4 subtrações

1. Interface gráfica do usuário, Texto, Aplicativo

   Descrição gerada automaticamente  
   R: subtrações
2. Interface gráfica do usuário, Texto, Aplicativo

   Descrição gerada automaticamente  
   R: subtrações
3. Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

   Descrição gerada automaticamente  
   R: 3 subtrações
4. Interface gráfica do usuário, Texto

   Descrição gerada automaticamente  
   R: 3 subtrações

## Estrutura de repetição dupla

1. Interface gráfica do usuário, Texto, Aplicativo

   Descrição gerada automaticamente  
   R: subtrações

## Estrutura de repetição com custo logarítmico

1. Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

   Descrição gerada automaticamente  
   R: Como são realizadas divisões sucessivas, o código realiza multiplicações

## Mais exercícios resolvidos

1. Diagrama, Texto

   Descrição gerada automaticamente com confiança média

…  
i = **0**;  
**while** (i < n){

i++;

a--; b--; c--;  
}  
**for** (i = **0**; i < n; i++){

**for** (j = **0**; j < n; j++){

a--; b--;  
}

}

…

i = 0;

while (i < n){

i++;

a--; b--; c--; d--; e--;

}

for (i = 0; i < n; i++){

for (j = 0; j < n; j++){

for (k = 0; k < n; k++, a--, b--, c--, d--);

}

}

…

i = 1;

while (i < n){

i\*= 2;

a--;

}

for (i = 0; i < n; i++){

a--;

}

…

for (i = 0; i < n; i++){

for (j = 0; j < n; j++){

for (k = 0; k < n; k++, a--, b--);

}

}

a--; b--; c--; d--; e--;

…

for (i = 0; i < n; i++){

for (j = 0; j < n; j++){

a--; b--;

for (k = 0; k < n; k++){

for (l = 0; l < n; l++, c--; d--);

}

}

if (i % 2 == 0) e--;

}

…

i = 1;

while (i < n){

i\*= 2;

a--; b--; c--; d--; e--; f--;

}

# Funções de Complexidade

1. Texto

   Descrição gerada automaticamente  
   Nesse código, a operação relevante é a comparação entre elementos de um array, que será executada vezes em todos os casos (visto que não há melhor ou pior caso).  
   Dessa forma, o algoritmo pode ser considerado como ótimo, visto que é preciso fazer sempre o mesmo número de operações para garantir a resposta.
2. Texto

   Descrição gerada automaticamente com confiança baixa  
   No melhor caso, no qual o elemento desejado está na primeira posição, são realizados comparações ente elementos do array.  
   Já na pior situação, são realizadas comparações. Isso ocorreria quando o elemento buscado está na última posição ou não está no array.
3. Texto

   Descrição gerada automaticamente  
   No melhor caso, quando o elemento desejado está na posição [(esq+dir)/2], a função de complexidade da pesquisa é .  
   Já, no pior caso, a função é , visto que são feitas divisões sucessivas do escopo em cada iteração. Esse caso acontece quando o elemento ou está na primeira/ultima posição do array ou não está contido.
4. Uma imagem contendo Diagrama

   Descrição gerada automaticamente  
   Como o laço se repete vezes, e cada loop executa um *swap* que realiza 3 movimentações, são feitas então movimentações de registros.
5. Texto

   Descrição gerada automaticamente com confiança baixa  
   As alterações estão marcadas de amarelo.

int mov = 0;

for (int i = 0; i < (n - 1); i++) {

int menor = i;

for (int j = (i + 1); j < n; j++){

if (array[menor] > array[j]){

menor = j;

}

}

swap(menor, i);

mov += 3;

}  
printf(“Teoria: ” + (3\*n - 3));  
printf(“Prática: ” + mov);

1. Texto

   Descrição gerada automaticamente  
   Como as comparações desejadas estão sendo feitas no *IF*, com o laço externo se repetindo vezes e o laço interno se repetindo vezes, temos que:

# Notação Ɵ

1. Interface gráfica do usuário, Texto, Aplicativo

   Descrição gerada automaticamente  
   No melhor caso, o código realiza , ou seja, . Já no pior caso, são realizadas , logo, .
2. Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

   Descrição gerada automaticamente  
   R: 3 subtrações.
3. Interface gráfica do usuário, Texto, Aplicativo

   Descrição gerada automaticamente  
   R: 4 subtrações.
4. Interface gráfica do usuário, Texto, Aplicativo

   Descrição gerada automaticamente  
   R: 8 subtrações
5. Interface gráfica do usuário, Texto

   Descrição gerada automaticamente  
   R: 12 subtrações
6. Interface gráfica do usuário, Texto, Aplicativo

   Descrição gerada automaticamente  
   R: multiplicações
7. Interface gráfica do usuário, Texto, Aplicativo

   Descrição gerada automaticamente  
   R: multiplicações
8. Interface gráfica do usuário, Texto, Aplicativo

   Descrição gerada automaticamente  
   R: multiplicações.
9. Texto

   Descrição gerada automaticamente  
   R: multiplicações.
10. Interface gráfica do usuário, Texto, Aplicativo

    Descrição gerada automaticamente  
    R: multiplicações.
11. Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

    Descrição gerada automaticamente  
    R: multiplicações.
12. Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

    Descrição gerada automaticamente  
    R: multiplicações.
13. Interface gráfica do usuário, Texto

    Descrição gerada automaticamente  
    R: multiplicações.