



Universidade de Ribeirão Preto
Campus Ribeirão Preto - Campus Guarujá

Campus Ribeirão Preto	Nota
Prova	6,8
Outras Avaliações	3,0
Média	9,8

Código ~~no~~ 823815

Aluno Márcio Gabriel Gomes dos Menezes

Disciplina Estadística de Datos I

Código EV922/A

Curso Engenharia de Computação

Sala/Bloco *28A*

Professor Yasi Ferreira Neto

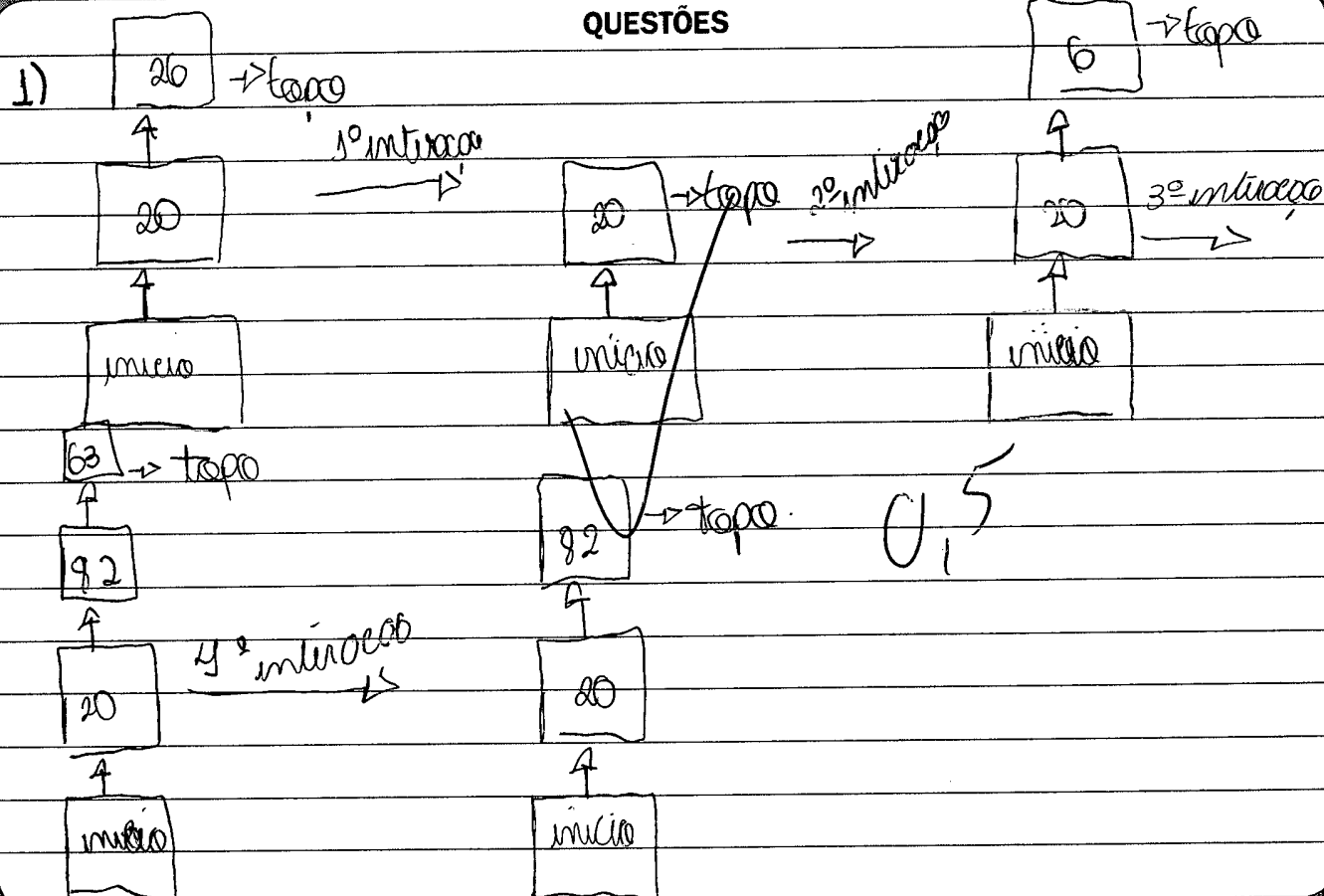
Início da Prova

Término da Prova

Data 05 / 12 / 18

INSTRUÇÕES

QUESTÕES



2) Struct Pilha {

struct Pilha * Prox;

};

typedef struct pilha ordema;

ordema * inicio = (ordema *) malloc(sizeof(ordema));

inicio->prox = null;

void Empilhar(ordema * inicio) {

ordema * novo = (ordema *) malloc(sizeof(ordema));

if (inicio->prox == null) {

inicio->prox = novo;

}

ordema * tmp;

tmp = inicio->prox;

while (tmp->prox != null) {

tmp = tmp->prox;

}

tmp->prox = novo;

novo->prox = null;

return;

}

3)

Struct Sort {

struct Sort * prox, * ant;

int elemento;

}

typedef Struct Sort ordema;

ordema * inicio = (ordema *) malloc(sizeof(ordema));

void Selection-Sort(ordema * inicio) {

int i, j;

ordema * tmp, * tmp2 = prox, * memora, * tmp2;

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

tmp = inicio->prox;

memora = tmp;

Código 823815

Aluno

maíseo gabriel jomeahen mangali

Disciplina

Estrutura de Dados I

Código

FV9221A

```

if (tmp->proxima == menor -> prox;
while (tmp->proxima != NULL) {
    if (tmp->proxima->elemento >= menor -> elemento) {
        tmp->proxima = tmp->proxima->prox;
    }
    if (tmp->proxima->elemento <= menor -> elemento) {
        menor = tmp->proxima;
        tmp->proxima = tmp->proxima->prox;
    }
}
if (menor != tmp) {
    tmp = tmp2;
    tmp = menor; // realocação de proximo elemento q menor
    menor = tmp2; // substituição de nó atual substituindo
    tmp = tmp->prox; // vála de variáveis
    menor = tmp;
    tmp->proxima = menor -> prox;
    j++;
}
if (menor == tmp) {
    tmp = tmp->prox;
    menor = tmp;
    tmp->proxima = menor -> prox;
    j++;
}
}
return;
}
    
```

4) Insert Sort

```
void Insert Sort (*pior, *out;  
int elemento;
```

```
{  
    typedef struct Sort node;
```

```
ordena * inicio = (ordena *) malloc(sizeof(ordena));
```

```
void Quick Sort (ordena * inicio) {
```

```
ordena * pivot, *tmp-esquerda, *tmp-direita; menor, maior;  
bool direito, esquerdo;
```

```
pivot = inicio -> piora -> direita; // escolhe um pivot
```

```
int i, j;
```

```
for (i = 0; i < j; i++) {
```

```
    // dividir a direita
```

```
    tmp-direita = pivot -> piora;
```

```
    menor = pivot;
```

```
    while (tmp-direita != null) {
```

```
        if (tmp-direita -> elemento >= menor -> elemento) {
```

```
            tmp-direita = tmp-direita -> piora;
```

```
        }
```

```
        if (tmp-direita -> elemento <= menor -> elemento) {
```

```
            menor = tmp-direita;
```

```
            tmp-direita = tmp-direita -> piora;
```

```
        }
```

```
    if (menor != pivot) {
```

```
        ordena * tmp2;
```

```
        tmp2 = pivot; // não estou satisfazendo os valores
```

```
        pivot = menor; // de pivot e menor menor
```

```
        menor = tmp2; // simbolizando a troca
```

```
        direita = false;
```

```
        j++;
```

```
    } // -- pivot
```

```
    if (menor == pivot) {
```

```
        direita = false;
```

```
    }
```

Campus Ribeirão Preto	Nota
Prova	
Outras Avaliações	
Média	

Código 823815 Aluno Márcio Gabriel J. Mengali
 Disciplina Estrutura de Dados I Código FV922A
 Curso Engenharia de Computação Sala/Bloco 28A
 Professor José Leandro Neto
 Início da Prova Término da Prova Data 05/12/18

INSTRUÇÕES

QUESTÕES

Continuação questão 4

```

//ordenar a esquerda
tmp-esquerda = P[ant] -> ont;
maior = P[ant];
while (tmp-esquerda != inicio) {
  if (tmp-esquerda -> elemento <= maior -> elemento) {
    tmp-esquerda = tmp-esquerda -> ont;
  }
  if (tmp-esquerda -> elemento > maior -> elemento) {
    maior = tmp-esquerda;
    tmp-esquerda = tmp-esquerda -> ont;
  }
}

if (maior != P[ant]) {
  P[ant] * tmp3;

```

$Emp3 = Pivot;$
 $Pivot = maior;$ // não estou modificando os valores de
 $maior = Emp3;$ // estou apenas simbolizando o troco
 $esquerda = true$ de posição.

$j++;$

$if(maior == pivot) \{$
 $esquerda = false$
 $\}$

$plena(ordenar, inicio) = // Função criada por mim e$
 $if(direta == true || esquerda == true) \{$ // as posições anteriores estão
 $j++;$ corretas.

$if(direta == false \& \& esquerda == false) \{$
 $pivot = pivot \rightarrow prox;$

$if(cheanca == true) \{$
 $Break$

$\}$

$return;$
 $\}$

5)

Insertion sort:

$q \quad p$
 $inicio \rightarrow 9 \rightarrow 3 \rightarrow 4 \rightarrow 12 \rightarrow 1 \rightarrow 2 \rightarrow 11 // q = inicio \rightarrow prox;$
 $null \quad \downarrow \quad null \quad p = q \rightarrow prox;$

$p \quad q$
 $inicio \rightarrow 3 \rightarrow 8 \rightarrow 4 \rightarrow 12 \rightarrow 1 \rightarrow 2 \rightarrow 11 \downarrow$
 $null \quad \downarrow \quad null$

$\Phi \quad \Phi$
 $inicio \rightarrow 3 \rightarrow 4 \rightarrow 8 \rightarrow 12 \rightarrow 1 \rightarrow 2 \rightarrow 11 \downarrow$
 $null \quad \downarrow \quad null$

$inicio \rightarrow 1 \rightarrow 3 \rightarrow 4 \rightarrow 8 \rightarrow 12 \rightarrow 2 \rightarrow 11 \downarrow$
 $null$

Código 823815

Aluno Márcio Gabriel J. Mengali

Disciplina Estrutura de Dados I

Código FV922A

Inicio $\rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 8 \rightarrow 12 \rightarrow 17$
null null

Inicio $\rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 8 \rightarrow 11 \rightarrow 12$
null null

selection Sort:

$8 \rightarrow 3 \rightarrow 4 \rightarrow 12 \rightarrow 1 \rightarrow 2 \rightarrow 11$
 $1 \rightarrow 8 \rightarrow 3 \rightarrow 4 \rightarrow 12 \rightarrow 2 \rightarrow 11$
 $1 \rightarrow 2 \rightarrow 8 \rightarrow 3 \rightarrow 4 \rightarrow 12 \rightarrow 11$
 $1 \rightarrow 2 \rightarrow 3 \rightarrow 8 \rightarrow 4 \rightarrow 12 \rightarrow 11$
 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 8 \rightarrow 12 \rightarrow 11$
 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 8 \rightarrow 11 \rightarrow 12$

X 0.8

Bubble Sort:

$8 \rightarrow 3 \rightarrow 4 \rightarrow 12 \rightarrow 1 \rightarrow 2 \rightarrow 11$
 $3 \rightarrow 8 \rightarrow 4 \rightarrow 12 \rightarrow 1 \rightarrow 2 \rightarrow 11$
 $3 \rightarrow 4 \rightarrow 8 \rightarrow 12 \rightarrow 1 \rightarrow 2 \rightarrow 11$
 $3 \rightarrow 4 \rightarrow 8 \rightarrow 12 \rightarrow 2 \rightarrow 1 \rightarrow 11$
 $3 \rightarrow 4 \rightarrow 12 \rightarrow 8 \rightarrow 2 \rightarrow 1 \rightarrow 11$
 $3 \rightarrow 4 \rightarrow 1 \rightarrow 2 \rightarrow 8 \rightarrow 11 \rightarrow 12$
 $3 \rightarrow 1 \rightarrow 4 \rightarrow 2 \rightarrow 8 \rightarrow 11 \rightarrow 12$
 $3 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 11 \rightarrow 12$
 $1 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 11 \rightarrow 12$
 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 8 \rightarrow 11 \rightarrow 12$

✓

Quick Sort:

Pivot

8 → 3 → 4 → 12 → 1 → 2 → 11

8 → 3 → 4 → 1 → 12 → 2 → 11

8 → 3 → 4 → 1 → 2 → 12 → 11

9 →

Pivot

8 → 3 → 4 → 1 → 2 → 11 → 12

1 →

Pivot

1 → 3 → 4 → 8 → 2 → 11 → 12

1 → 3 → 4 → 2 → 8 → 11 → 12

1 → 3 → 2 → 4 → 8 → 11 → 12

1 → 2 → 3 → 4 → 8 → 11 → 12

6)

*tmp1, *tmp2;

tmp1 = temp → ont;

tmp2 = aux → prox;

tmp1 → prox = aux;

aux → ont = tmp1;

aux → prox = tmp2;

temp → ont = aux;

temp → prox = tmp2;

tmp2 → ont = temp;