Universidade Federal da Fronteira Sul - UFFS
Campus Chapecó
Ciência da Computação
Estrutura de Dados

Instruções

1. Coloque apenas a sua MATRÍCULA na folha resposta.
2. Aparelhos eletrônicos desligados (ou no modo silencioso).
3. Responda TODAS questões na folha resposta, enumerando as mesmas na ordem que melhor lhe convier.
4. As questões podem ser resolvidas a lápis porém o professor se reserva a não aceitar reclamações oriundas da correção das questões.
5. Consulta permitida apenas à cola oficial.
6. Coloque, na folha resposta, o nome do professor da sua turma.

Avaliação 03

1. Um sistema produziu uma lista simples encadeada com id únicos de processos de impressão a serem enviados para a impressora (veja estrutura da lista abaixo - `sl`). Faça uma função que receba o primeiro elemento da lista e retorne uma fila representando a lista de identificadores utilizando a assinatura `queue *getQueue (sl *h)`, onde `h` representa o ponteiro para o primeiro elemento da lista e o retorno da função deve ser um ponteiro do tipo queue. 5 pt

```
// Lista simples
struct tsl{
  int id;
  struct tsl *next;};
typedef struct tsl sl;
sl *head;
```

```
// Fila
struct tq {
   int id;
   struct tq *next;};
typedef struct tq q;
typedef struct {
   q *head, *tail;
} queue;
```

2. Dados o seguinte vetor ordenado [4, 7, 9, 12, 14, 17, 21, 26, 30, 32, 34, 37, 43, 45, 49, 51, 53, 56] e a chave 45, utilizando o conceito de busca binária: (*a*) calcule e apresente quantas comparações serão feitas até a chave ser encontrada e os valores do índice do meio até o encontro da chave (por exemplo, se a chave fosse 14, você deveria imprimir: *2 comparações; meio: 9 e 4*, em seguida, dizer que a chave foi encontrada na *posição 4*); (*b*) apresente quantas comparações seriam feitas se a chave procurada não existisse e fosse menor que 4, e (*c*) se fosse maior que 56. 3 pt

3. A função abaixo foi lhe enviada para você identificar se ela é eficiente em tempo de execução ou não. Ela será eficiente se a sua complexidade for $\mathcal{O}(n)$ ou inferior. Assim, você deverá calcular a complexidade da mesma e indicar, baseado na complexidade encontrada (notação assintótica), se a função é eficiente ou não. 2 pt

```
void magicConversor (int *v, int n)
{
  int i=n;
  while (i>=1) {
    v[i]=converte(n); // função converte é O(1)
    i=i/2; }
}
```

BOA PROVA & BOA SORTE.