

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 duplamente encadeada com números únicos, ordenados em ordem crescente (veja estrutura da lista abaixo - ld).
Faça: (a) uma função que receba o sentinela da lista dupla e devolva a estrutura de uma fila (*queue*) em que os valores na fila sejam os mesmos na ordem da lista: `queue getQueue(sent st)` (3 pt); e (b) outra função que receba o sentinela da lista dupla e devolva a estrutura de uma pilha (*stack*) em que os valores na pilha sejam armazenados na ordem inversa deles na lista: `stack getStack(sent st)` (3 pt).

```
// Lista dupla
struct tld{
    int n;
    struct tld *next, *prev;};
typedef struct tld ld;
typedef struct {
    ld *head;
    ld *tail;
} sent;
```

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

```
//Pilha
struct ts {
    int n;
    struct ts *next;};
typedef struct ts s;
typedef struct {
    s *top;
} stack;
```

2. Utilize a seguinte lista de números inteiros [35, 20, 7, 9, 30, 25, 50, 45, 60] para criar uma árvore binária de busca ((a) - 1pt). Indique se a árvore resultante é balanceada ((b) - 1pt). Em seguida, esboce o funcionamento do método de percurso *pré-ordem* e mostre como ficaria a saída da árvore criada ((c) - 1pt).
3. A função abaixo (*Xmen*) recebe um vetor *v* de tamanho *n* e realiza um determinado processamento. Analise as instruções e apresente a equação que represente o número de vezes que cada instrução vai ser executada e depois, usando a notação O , o pior caso. Coloque $c_1 \times$ o número de vezes que a instrução executa, somando todos os cálculos (1 pt).

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