

Verificação de Aprendizagem 3

Rodrigo de Souza

27 de Agosto de 2018

1 Instruções

- Faça um cabeçalho (nome, disciplina, número da prova – VA 3, data).
- Você pode descrever código em Português, usar pseudocódigo ou mesmo usar linguagem C.
- Cada questão vale 2 pontos.

2 Enunciados

1. Escreva uma função recursiva que calcule a soma dos dígitos decimais de um inteiro estritamente positivo n . A soma dos dígitos de 132, por exemplo, é 6. Estime o número de chamadas recursivas de seu algoritmo.
2. As três funções a seguir¹ constituem o Heapsort para colocar um vetor em ordem crescente. A primeira etapa é transformar o vetor em um MAX-HEAP. Mostre como essas funções podem ser adaptadas a fim de colocar o vetor em ordem decrescente. Para isso, transforme, em primeiro lugar, o vetor em um MIN-HEAP.

```
// Rearranja um vetor v[1..m] de modo a
// transformá-lo em heap.

static void
constroiHeap (int m, int v[])
{
    for (int k = 1; k < m; ++k) {
        // v[1..k] é um heap
        int f = k+1;
        while (f > 1 && v[f/2] < v[f]) { // 5
            troca (v[f/2], v[f]);      // 6
            f /= 2;                     // 7
        }
    }
}
```

¹Fonte: <https://www.ime.usp.br/~pf/algoritmos/aulas/hpsrt.html>

```

static void
peneira (int m, int v[]) {
    int f = 2;
    while (f <= m) {
        if (f < m && v[f] < v[f+1]) ++f;
        // f é o filho mais valioso de f/2
        if (v[f/2] >= v[f]) break;
        troca (v[f/2], v[f]);
        f *= 2;
    }
}

// Rearranja os elementos do vetor v[1..n]
// de modo que fiquem em ordem crescente.
void
heapsort (int n, int v[])
{
    constroiHeap (n, v);
    for (int m = n; m >= 2; --m) {
        troca (v[1], v[m]);
        peneira (m-1, v);
    }
}

```

3. Escreva uma função que recebe uma fila f e copia seu conteúdo em uma pilha p , inicialmente vazia. A sequência dos elementos em p , lendo-se da base ao topo, deve ser igual à sequência em f , lendo-se do último ao primeiro. Você não sabe como é a implementação de f , nem de p ; nem sabe o tamanho de f . Você só tem o direito de usar as operações insere/remove em fila, e empilha/desempilha em pilha. Você pode usar uma ou mais estruturas auxiliares de pilha e fila. Note que sua função pode manipular a fila f , mas ao final da execução, ela é igual à original.
4. Descreva uma função que recebe duas listas ligadas, com cabeça, cada uma representando uma sequência *crescente* de números inteiros, e constrói uma nova lista, contendo os inteiros de ambas as listas intercalados em ordem crescente (semelhante ao que se faz na etapa de intercalação do **Mergesort**). Essa nova lista deve aproveitar os nós das listas recebidas (que portanto deixam de existir individualmente).
5. Descreva um algoritmo recursivo, do tipo divisão-e-conquista, que transforma um vetor crescente $v[0..n-1]$ em uma árvore binária de busca *balanceada* (os valores do vetor são armazenados em um campo *conteudo* dos nós). Atenção ao estruturar seu algoritmo: que parâmetros ele recebe, o que devolve? Seu algoritmo deve ter complexidade $\mathcal{O}(n)$. Diga qual é a *altura da árvore resultante*.