



Instituto Federal de Educação, Ciência e Tecnologia de Brasília – Câmpus Taguatinga
Ciência da Computação – Estruturas de Dados e Algoritmos
Prova I – 1º/2017 – Algoritmos, Alocação Dinâmica, Ordenação e Busca
Prof. Daniel Saad Nogueira Nunes

Aluno: _____

Matrícula: _____

Data: 10 de maio de 2017

Duração da prova: 100 minutos

Questão 1 (1 ponto)

Crie uma função que aloque uma matriz de tipo ponto flutuante de precisão dupla de maneira dinâmica dados parâmetros n e m sendo respectivamente n o número de linhas e m o número de colunas desta matriz e retorne esta matriz alocada.

Questão 2 (3 pontos)

Suponha que tenhamos n cidades, identificadas por números de 0 a $n - 1$. As ligações entre a cidade i e a cidade j por meio de uma estrada é representada pelo elemento $M[i][j]$ de uma matriz, isto é, $M[i][j] = 1$ se existe uma estrada da cidade i à cidade j , caso contrário $M[i][j] = 0$. Note que se existe uma estrada de i a j , não necessariamente existe uma estrada de j a i , somente se $M[j][i]$ também possuir o valor 1. Observe também que o tamanho da matriz é $n \times n$. Dado esta matriz, e o tamanho n como parâmetro, escreva funções que imprimam:

- (a) (1.5 pontos) As cidades que não possuem saídas.
- (b) (1.5 pontos) As cidades cuja quantidade de entradas é igual a quantidade de saídas.

A assinatura dessas funções deve ser obrigatoriamente:

```
void <nomedafuncao> (int** m, size_t n);
```

Questão 3 (4 pontos)

De acordo com os métodos de ordenação:

- (a) (2 pontos) Implemente a função de **partition** do Quicksort, que classifica o vetor em três partes: partição dos elementos menores do que o pivô, pivô, e partição dos elementos maiores que o pivô. Sua função deve possuir a seguinte assinatura:

```
void partition(int* v, int l, int r);
```

onde v é o vetor a ser particionado, l é o início desse vetor e r o fim.

★ Certifique-se de assinar todas as folhas de resposta.

-
- (b) (2 pontos) Implemente o método de ordenação por inserção. Seu método deve possuir a seguinte assinatura:

```
void insertionsort(int* v, size_t n);
```

Questão 4 (4 pontos)

- (a) (2 pontos) Discorra sobre o porquê da busca binária ser eficiente, se comparada com a busca sequencial, quando aplicada em um vetor ordenado para buscar uma determinada chave.
- (b) (2 pontos) Implemente a busca binária sobre um vetor de inteiros ordenado de modo que retorne -1 se a chave k não encontra-se no vetor v ou a posição de k no vetor, caso contrário. Sua função deve ter a seguinte assinatura:

```
int bsearch(int*v, size_t n, int k);
```