



Instituto Federal de Educação, Ciência e Tecnologia de Brasília – Câmpus Taguatinga
Ciência da Computação – Estrutura de Dados e Algoritmos
Prova I – 1º/2018 – Ponteiros e Ordenação
Prof. Daniel Saad Nogueira Nunes

Aluno: _____

Matrícula: _____

Data: 18 de novembro de 2020

Duração: 120 minutos

Tabela de notas (uso exclusivo do professor)

| Questão | Pontos | Nota |
|---------|--------|------|
| 1 | 2½ | |
| 2 | 2½ | |
| 3 | 2½ | |
| 4 | 2½ | |
| Total | 10 | |

Observações

- Esta prova tem o total de 3 páginas (incluindo a capa) e 4 questões.
- O número total de pontos é 10.
- Certifique-se de assinar todas as folhas de resposta bem como a capa da prova.
- Leia atentamente todas as questões da prova. A interpretação do problema é crucial para o desenvolvimento correto da resposta.
- Resoluções sem justificativa não serão consideradas.
- É vedado o uso de equipamentos eletrônicos, como celulares, notebooks entre outros.
- A prova será **anulada** e medidas disciplinares serão tomadas para os alunos que “colarem” durante a avaliação.

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

Questão 1 (2½ pontos)

A ordenação por classificação sobre vetores inteiros contendo valores na faixa $[a, b]$ funciona da seguinte maneira:

- Crie um vetor c de tamanho $[b - a + 1]$.
- Defina $c[i - a]$ como o número de vezes que o número i aparece no vetor v .
- Utilize c para ordenar o vetor v ao inspecioná-lo da esquerda para direita e inserir elementos em v enquanto $c[i]$ é positivo. A cada elemento i inserido, $c[i]$ é decrementado.

Implemente a função `bucketsort(int* v, size_t size)`; que ordena um vetor de inteiros positivos de acordo com este método.

Questão 2 (2½ pontos)

Tome o seguinte tipo `aluno`:

```
typedef struct aluno{
    char nome[80];
    char matricula[40];
    int semestre;
    double renda_familiar;
} aluno;
```

Suponha que tenhamos um vetor de alunos de tamanho `n` ordenado pela matrícula de acordo com a ordem lexicográfica (ordem do dicionário). Faça um programa que efetue uma busca binária de uma matrícula e imprima a renda familiar deste aluno. Caso o aluno não seja encontrado, o programa deverá imprimir uma mensagem de erro indicando que o vetor não possui aquele aluno.

A sua função deverá possuir a seguinte assinatura:

```
void busca_matricula(aluno* vetor_aluno, size_t n, char* matricula);
```

OBS: é permitido utilizar a função `strcmp`.

Questão 3 (2½ pontos)

Crie uma função que leia caracteres enquanto um `\n` não for digitado e armazene-os em uma string dinâmica. A função deverá retornar a string lida e ela deverá terminar com o caractere de `\0`. Para efetuar esta tarefa você deverá utilizar funções de manipulação de memória como `malloc`, `calloc` ou `realloc`. Esta função a ser implementada deverá possuir a seguinte assinatura:

```
char* getline(void);
```

Questão 4 (2½ pontos)

De acordo com os métodos de ordenação vistos em sala de aula:

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

-
- (a) (1.25 pontos) Implemente a função `heapify(int* v, int raiz, size_t n);`, que restaure a propriedade de heap a partir do elemento `raiz` com uma heap de tamanho `n`.
- (b) (1.25 pontos) Implemente o `insertionsort(int* v, size_t n);` substituindo a busca sequencial, que encontra a posição de inserção do novo elemento, por uma busca binária.

A vingança nunca é plena, mata a
alma e a envenena

Seu Madruga