
Instituto Federal de Educação, Ciência e Tecnologia de Brasília – Câmpus Taguatinga
Ciência da Computação – Estrutura de Dados e Algoritmos
Prova II – 2°/2018 – Listas, Filas, Pilhas, Deques e Filas de Prioridade
Prof. Daniel Saad Nogueira Nunes

Aluno: _____
Matrícula: _____
Data: 18 de outubro de 2018

| Duração da prova: 100 minutos |
|---|

Tabela de notas (uso exclusivo do professor)

| Questão | Pontos | Nota |
|---|---|---|
| 1 | $2\frac{1}{2}$ | |
| 2 | 3 | |
| 3 | 3 | |
| 4 | $1\frac{1}{2}$ | |
| 5 | 2 | |
| Total | 12 | |

## Observações

- Esta prova tem o total de 3 páginas (incluindo a capa) e 5 questões.
- O número total de pontos é 12.
- 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)

Implemente o procedimento de filtragem em uma lista que, dado um predicado (função que recebe um dado qualquer e o avalia em verdadeiro ou falso), deixa na lista original somente os elementos cujo predicado é verdadeiro. Este procedimento deverá seguir a seguinte assinatura:

```
void list_filter(list_t* list,int (*fn)(void* data));
```

Utilize as definições dos tipos presentes na biblioteca da disciplina.

## Questão 2 (3 pontos)

Com relação a pilhas e filhas e utilizando as definições de tipos presentes na biblioteca da disciplina:

(a) (1½ pontos) Implemente a inserção em uma pilha. A seguinte assinatura deverá ser utilizada:

```
void stack_push(stack_t* pilha,void* data);
```

(b) (1½ pontos) Implemente a remoção em uma fila. A seguinte assinatura deverá ser utilizada:

```
void queue_pop(queue_t* fila);
```

## Questão 3 (3 pontos)

Suponha que você tenha $N$ inteiros em uma pilha. Seja $K$ um parâmetro inteiro. Você deverá retirar $K$ elementos da pilha de modo a obter a maior soma. Mas existe um truque, em qualquer momento, você pode transformar a pilha em uma fila de modo que o fundo da pilha corresponda à frente da pilha e, a partir daí, os elementos podem ser retirados da frente da fila. Este é um processo irreversível, isto é, uma vez que a transformação em fila seja efetuada, é impossível voltar atrás.

Elabore um algoritmo que, ao receber a pilha original, retorne a maior soma da retirada dos $K$ elementos obedecendo as restrições do problema.

Por exemplo, se os elementos que estão na pilha são: $(10, 9, 1, 2, 3, 4, 5, 6, 7, 8)$, sendo que 10 está no topo da pilha, e o parâmetro $K = 5$, a melhor solução seria:

- Retirar 10 e 9 da pilha.
- Transformar a pilha em uma fila.
- Retirar 8, 7 e 6 da fila.

Esta solução fornece soma 40.

Seu algoritmo deverá calcular a soma máxima e retorná-la, considerando a seguinte assinatura:

```
int soma_maxima(stack_t* pilha,int k);
```

Considere para esta questão que todas as funções da biblioteca da disciplina já estão implementadas.

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

## Questão 4 (1½ pontos)

Implemente a função `take` que recebe uma lista e um inteiro $k$ e retorna uma nova lista com os $k$ primeiros elementos da lista original. Caso $k = 0$, a lista vazia deverá ser retornada. Esta função deverá possuir a seguinte assinatura:

```
list_t* take(list_t* list,int k);
```

## Questão 5 (2 pontos)

Suponha uma lista de nomes em ordem crescente (considerando a ordem do dicionário). Escreva uma função que recebe um novo nome e o insere nesta lista mantendo-a ordenada. Sua função deverá possuir a seguinte assinatura:

```
void insert_sorted(list_t* l,char* novo_nome);
```

Utilize as definições dos tipos presentes na biblioteca da disciplina.

Obs: você pode utilizar a função `strcmp` para comparar duas strings. Ela recebe as strings a serem comparadas e retorna:

- Um número negativo, caso a primeira string seja menor do que a segunda.
- 0, caso as duas strings sejam iguais,
- Um número positivo, caso a primeira string seja maior do que a segunda.

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