

Atividade prática
Professor Wladimir A. Tavares
14/11/2024

1 O maior menor do que a média

Imagine que nós temos uma lista que contém números inteiros. Por exemplo,

	0	1	2	3	4	5	6	7	8	10
V	5	1	3	10	8	13	6	12	4	16

A tarefa consiste em encontrar o maior elemento menor do que a média dos números da lista. No exemplo acima, isso nos daria 6 (porque a média é 7.8).
Implemente a seguinte função:

```
int maior_menor_media(int L[], int N){  
}
```

2 Ímpares em ordem crescente

Imagine que nós temos uma lista que contém números inteiros. Por exemplo,

	0	1	2	3	4	5	6	7	8	10
V	2	1	8	4	5	6	16	7	9	12

A tarefa consiste em verificar se os números ímpares da lista aparecem em ordem crescente. Os números ímpares da lista são [1,5,7,9].

No exemplo acima, isso nos daria Sim

Dica: Utilize um apontador ultimo_impar. Inicialmente, ultimo_impar aponta para o primeiro ímpar. Percorra a lista e toda vez que encontrar um número ímpar, cheque se os números estão em ordem crescente e atualize o ultimo_impar.

Implemente a seguinte função:

```
int impares_ordem_crescente(int L[], int N){  
}
```

3 Remover caracteres duplicados consecutivos.

Imagine que nós temos uma lista de caracteres V. Por exemplo,

	0	1	2	3	4	5	6	7	8	10
V	a	a	a	b	b	b	c	c	d	d

Remova os caracteres duplicados da lista de caracteres V

	0	1	2	3	4	5	6	7	8	10
V	a	b	c	d	\0	\0	\0	\0	\0	\0

Dica: Percorra a string e copie cada caractere apenas se ele for diferente do anterior.
Implemente a seguinte função:

```
void remova_duplicados_consecutivos(char L[], int N){  
}
```

4 Número de ocorrência de um elemento

Considere o seguinte problema: Você recebe uma lista de inteiros ordenada L de tamanho n com repetições e um inteiro k , se o inteiro k aparece no vetor devolva o número de ocorrência do elemento k na lista, caso contrário devolva -1.

	0	1	2	3	4	5	6	7	8
L	1	2	2	3	3	4	5	6	7

Quando $k=2$, o número de ocorrências do número de 2 é 2..

1. Adapte o algoritmo de busca linear para resolver esse problema.

```
int num_ocorrencia_linear(int L[], int N, int k){  
}
```

2. Adapte o algoritmo de jump search com passo \sqrt{n} para resolver esse problema.

```
int numero_ocorrencia_jump(int L[], int N, int k){  
}
```

Dica: Adapte o algoritmo para encontrar a primeira e a última ocorrência de k na lista L .

5 Sequência Bitônica

Uma sequência bitônica é uma sequência de números que, em algum ponto, muda de crescente para decrescente. Dado uma lista L de tamanho n que representa uma sequência bitônica, seu objetivo é encontrar o pico da sequência — ou seja, o elemento de maior valor da sequência.

	0	1	2	3	4	5	6	7	8
L	0	1	4	9	16	12	10	5	3

Na lista acima, o pico é o valor 16 que ocorre na posição 4.

1. Adapte o algoritmo de linear para resolver esse problema.

```
int pico_bitonica_linear(int L[], int N){  
}
```

2. Adapte o algoritmo de busca binária para resolver esse problema.

```
int pico_bitonica_binaria(int L[], int N){  
}
```