Universidade do Estado do Amazonas
Escola Superior de Tecnologia
Algoritmos e Estrutura de Dados II (Code: ESTCMP012)

12 de dezembro de 22
Tempo: 08 dias

Trabalho Prático I

Especificação do Trabalho

1 Problema

Ordenação de dados é um problema clássico da Ciência da Computação. A seguir, será descrito o algoritmo de ordenação chamado *vira-vira*.
O algoritmo *vira-vira* é um algoritmo de ordenação no qual a única operação permitida é *virar* uma extremidade da lista (coleção) de dados. Diferentemente dos algoritmos de ordenação tradicionais, que tentam ordenar com o menor número de comparações, a ordenação deste algoritmo tenta executar a sequência com o menor número possível de reversões.
A lógica por trás da ordenação do algoritmo *vira-vira* é semelhante à ordenação por seleção. Em cada iteração, nós encontramos o elemento máximo (maior) na sequência e o colocamos no final e reduzimos o tamanho da sequência de um.
Por exemplo, dada uma L de inteiros, deve-se escrever uma função `vira(L, i)` que inverte a ordem dos primeiros elementos `i` na lista `L`. O algoritmo usa **apenas** essa função para fazer as alterações na lista L.

2 Exemplo

A seguir, é apresentado um exemplo de ordenação usando o algoritmo *vira-vira*:
Considere a lista inicial L = [30, 50, 20, 10, 70, 60, 40]

Primeira virada : [**30, 50, 20, 10, 70**, 60, 40]
Depois da primeira virada : [**70, 10, 20, 50, 30**, 60, 40]

Segunda virada : [**70, 10, 20, 50, 30, 60, 40**]
Depois da segunda virada : [**40, 60, 30, 50, 20, 10, 70**]

Terceira virada : [**40, 60**, 30, 50, 20, 10, 70]
Depois da terceira virada : [**60, 40**, 30, 50, 20, 10, 70]

Quarta virada : [**60, 40, 30, 50, 20, 10**, 70]
Depois da quarta virada : [**10, 20, 50, 30, 40, 60**, 70]

Quinta virada : [**10, 20, 50**, 30, 40, 60, 70]
Depois da quinta virada : [**50, 20, 10**, 30, 40, 60, 70]

Sexta virada : [**50, 20, 10, 30, 40**, 60, 70]
Depois da sexta virada : [**40, 30, 10, 20, 50**, 60, 70]

Sétima virada : [**40, 30, 10, 20**, 50, 60, 70]
Depois da sétima virada : [**20, 10, 30, 40**, 50, 60, 70]

Oitava virada : [**20, 10**, 30, 40, 50, 60, 70]
Depois da oitiva virada : [**10, 20**, 30, 40, 50, 60, 70]

3 Entrada de Dados

O arquivo de entrada terá apenas uma linha. A linha terá o conjunto de valores inteiros que representarão os elementos. A entrada é sempre bem formatada.

4 Saída de Dados

O programa deverá imprimir a saída dos elementos de maneira ordenada em uma única linha.

5 Exemplo

5.1 Exemplo - Entrada

```
30, 50, 20, 10, 70, 60, 40
```

5.2 Exemplo - Saída

```
10, 20, 30, 40, 50, 60, 70
```

6 Requisitos

O programa desenvolvido deverá ter duas funções:
A função `acharMaior(L , t)` que deverá receber como parâmetro a `lista L` e o o `tamanho t` da lista L a ser considerado. Essa função deverá retornar o `índice i` que indica o maior elemento de L no intervalo até n.
A função `vira (L, i)` que deverá receber como parâmetro a `lista L` e o `índice i` da lista L que será usado como intervalo para virar os valores. A função não deverá ter retorno.
As funções `acharMaior(L , t)` e `vira (L, i)` são obrigatórias e devem seguir o padrão descrito no trabalho.

7 Prazos

- Início: 13/122022 às 10:00 horas (horário do servidor).
- Encerramento: 20/12/2022 às 10:00 horas (horário do servidor).

8 Regras gerais

- Lembrem-se, a entrada de dados é feita via `input` e a saída via `print`.
- Não é necessário mostrar mensagem para entrada de dados. A saída deve apresentar apenas a resposta sem mensagem adicional.
- Na construção do seu programa você deve usar apenas os conceitos aprendidos em sala de aula. Respostas que utilizem bibliotecas prontas não serão consideradas.
- Em caso de plágio, **todos** os envolvidos receberão **nota zero**!
- Na execução do seu programa no `run.codes`, existem casos de testes que vão além dos exemplos mostrados a seguir. Esses casos de teste não serão revelados. Pense em exemplos de entradas e saídas que podem acontecer e melhore o seu código para capturá-las.
- Cada aluno deve ter **apenas uma conta**. Se houver duplicidade de resposta do mesmo aluno, será atribuída **nota zero**!
- Esse é um trabalho prático e o resultado fará parte no cômputo da nota da disciplina.
- A interpretação do problema faz parte da avaliação. Você deve fazer o problema sozinho. Evite discutir a solução com outras pessoas, **especialmente com outros colegas de sala de aula**.
- A legibilidade do código-fonte será também considerada na avaliação.
- Se for necessário, o professor poderá arguir a defesa do trabalho submetido ao aluno. Essa arguição deverá ocorrer de forma não presencial (*online*).
- Os alunos que não tiverem acesso ao computador em casa deverão utilizar os laboratórios de informática disponibilizados na Escola Superior de Tecnologia (EST).
