

=====

SSC0603 – ED1 – Prof. Fernando Osório – ICMC / USP 2021-2
Prof. Fernando Osório – Estrutura de Dados 1 (Eng.Comp.)
Run.Codes: Trabalho Prático - TP-02 Versão TP-02 v.01
Trabalho: Dados3D - Listas Encadeadas x Matrizes Esparsas

=====

TP02 - Dados3D em Listas

**** Definição Geral do Trabalho:** TADs LDES (simples) ou TAD LDED (dupla), podendo considerar uma estrutura do tipo matrizes esparsas de dados 3D (coordenadas X, Y, Z, com 3 índices), e incluindo TADs em C (sugere-se os TADs do Backes e de seu livro), ou em C++ ou Python, (*) Aceito em Python e C++ mas tem que usar LISTAS ENCADEADAS com ALOCAÇÃO DINÂMICA na Ling. Python 3 com rotinas usando comandos como “self.data”, “self.ref”, “self.next”

Este trabalho é relacionado ao uso de **listas encadeadas dinâmicas** (simples ou duplamente encadeadas) e que irão ser usadas para armazenar os dados 3D em matrizes esparsas com dados ordenados e com consulta “rápida” priorizando alguma das coordenadas. O programa também irá apresentar o desempenho na consulta aos dados destas diferentes listas com dados 3D esparsos. Serão criadas 5 listas encadeadas com os MESMOS dados nas 5 listas, o que vai permitir depois comparar o “custo” de acessar os dados:

1. Lista ordenada em X, depois em Y, e depois em Z (em caso de empate: X -> Y -> Z)
2. Lista ordenada em Y, depois em X, e depois em Z (em caso de empate: Y -> X -> Z)
3. Lista ordenada em Z, depois em X, e depois em Y (em caso de empate: Z -> X -> Y)
4. Lista não ordenada com inserção no início
5. Lista não ordenada com inserção no final

Deve dar para notar que os custos de inserção (nro. de nodos visitados) nas 5 listas serão diferentes e os custos de consulta também serão diferentes, conforme o dado que vamos querer acessar/procurar.

Os dados estão armazenados em um arquivo (“dados3d.txt”) que será lido e então esses dados serão colocados nas listas, são inseridos os mesmos dados nas 5 listas, ou seja, haverá 5 cópias dos dados espalhados nestas listas. Os dados manipulados são chamados de point-clouds 3D (XYZ+CorRGB+ID). Os dados armazenados devem guardar informações sobre:

- XYZ: A coordenada do ponto X,Y,Z (3 valores do tipo float)
- RGB: A cor do ponto com valores de cor em R, G, B (Red, Green, Blue) com valores inteiros. Os valores de cor variam de 0 a 255 cada um deles (R,G,B são 3 valores inteiros positivos)
- ID: Um valor que identifica o “objeto”, ou seja, um identificador, que é um valor entre 0 e 1024. Os pontos 3D podem fazer parte de diferentes objetos, e cada objeto possui um identificador próprio (um valor inteiro entre 0 e 1024).

Exemplo de arquivo de entrada com dados 3D: “dados3d.txt”

```
10.0 1.0 1.0 255 0 0 1
5.00 1.0 1.0 255 0 0 1
1.00 1.0 1.0 255 0 0 1
3.00 1.0 2.0 255 0 0 1
8.00 1.0 2.0 255 0 0 1
15.0 1.0 2.0 255 0 0 1
0 0 0 0 0 0 -1
```

Note que o arquivo termina pela linha “0 0 0 0 0 0 -1” que não deve ser inserida nas listas. O valor do identificador possui um valor inválido (-1), indicando que não tem mais dados.

O programa começa lendo os dados e inserindo os dados do arquivo nas 5 listas, conforme descrito acima. Lembrando: cada uma das 5 listas deve conter todos os dados, ou seja, são 5 listas contendo cada uma delas todos dados lidos do arquivo (sim, os dados são replicados). Após lidos os dados o programa entra em modo “consulta”, recebendo comandos do teclado que permitem obter respostas exibidas na tela. Por exemplo: se o usuário digitar o comando “lx”, então devem ser exibidos os dados da lista ordenada por X (ordem de prioridade X, depois Y, depois Z).

Os comandos digitados pelo teclado e que devem ser aceitos pelo programa são:
(sempre digitados em maiúsculas)

LX – Listar todos dados da lista ordenada por X (X depois Y depois Z)

Formato: X, Y, Z, R, G, B, ID [dados separados por vírgula e espaço, float com 5 casas]

Exemplo: 1.00000, 1.00000, 1.00000, 255, 0, 0, 1 [XYZ=(1.0;1.0;1.0); RGB=(255,0,0), ID=1]

LY - Listar todos dados da lista ordenada por Y (Y depois X depois Z)

LZ - Listar todos dados da lista ordenada por Z (Z depois X depois Y)

LI - Listar todos dados da lista com inserção NÃO ordenada, inserindo dados no início da lista

LF - Listar todos dados da lista com inserção NÃO ordenada, inserindo dados no final da lista

TOT – Exibir o total de nodos acessados nas inserções nas listas LX,LI,LF. Considerando o exemplo “dados3d.txt” citado acima seria exibido: LX:15 LI:6 LF:21

PZ – Listar o dado mais próximo em Z contido na lista ordenada por **Z**, ou seja, mostrar o dado com o menor valor de Z e o total de nodos visitados para achar este dado. Exemplo:
1.00000 1.00000 1.00000 TOTAL:1

PI - Listar o dado mais próximo em Z, contido na lista não ordenada com inserção no **Início**, ou seja, mostrar o dado com o menor valor de Z e o total de nodos visitados para achar este dado. Exemplo: 1.00000 1.00000 1.00000 TOTAL:6

ID – Ler do teclado um número inteiro representando o ID procurado e exibir apenas os dados que possuem o ID informado. Exibir no formato igual ao das opções LX,LY,LZ,LI e LF, porém aplicando um filtro que exibe apenas os dados com o ID solicitado. Consultar a lista ordenada por X (LX – X depois Y depois Z) para exibir os dados. Exemplo: ID 5

1.11000, 2.22000, 3.33000, 255, 255, 255, 5

4.44000, 5.55000, 6.66000, 0, 0, 0, 5

CUT – Ler do teclado 3 números com as coordenadas X,Y,Z mínimas e 3 números com as coordenadas X,Y,Z máximas, definindo um “bounding box” (retângulo de recorte) dos dados 3D. Considerando a lista LX (ordem em X depois Y e depois Z, exibir na tela apenas os dados que ficam dentro do retângulo definido (incluindo os limites max e min). Exemplo:

Entrada de dados pelo teclado: CUT 0.0 0.0 0.0 10.0 10.0 10.0

Vai exibir na tela os seguintes dados: (considerando o exemplo acima do “dados3d.txt”)

1.00000, 1.00000, 1.00000, 255, 0, 0, 1

3.00000, 1.00000, 2.00000, 255, 0, 0, 1

5.00000, 1.00000, 1.00000, 255, 0, 0, 1

8.00000, 1.00000, 2.00000, 255, 0, 0, 1

10.00000, 1.00000, 1.00000, 255, 0, 0, 1

Você pode implementar uma “opção oculta” (OPCIONAL), que NÃO será usada no RUN.CODES para gravar em disco os dados 3D, que depois poderão ser exibidos por um visualizador 3D. O professor irá explicar como salvar os dados e como visualizar em 3D. Comando teclado: SAVE

**** Exemplos de Dados:**

Arquivo "dados3d.txt":

```
10.0 1.0 1.0 255 0 0 1
5.00 1.0 1.0 255 0 0 1
1.00 1.0 1.0 255 0 0 1
3.00 1.0 2.0 255 0 0 1
8.00 1.0 2.0 255 0 0 1
15.0 1.0 2.0 255 0 0 1
0 0 0 0 0 0 -1
```

Entrada: (teclado)

LX

Saída: (tela)

```
1.00000, 1.00000, 1.00000, 255, 0, 0, 1
3.00000, 1.00000, 2.00000, 255, 0, 0, 1
5.00000, 1.00000, 1.00000, 255, 0, 0, 1
8.00000, 1.00000, 2.00000, 255, 0, 0, 1
10.00000, 1.00000, 1.00000, 255, 0, 0, 1
15.00000, 1.00000, 2.00000, 255, 0, 0, 1
```

Entrada: (teclado)

LY

Saída: (tela)

```
1.00000, 1.00000, 1.00000, 255, 0, 0, 1
3.00000, 1.00000, 2.00000, 255, 0, 0, 1
5.00000, 1.00000, 1.00000, 255, 0, 0, 1
8.00000, 1.00000, 2.00000, 255, 0, 0, 1
10.00000, 1.00000, 1.00000, 255, 0, 0, 1
15.00000, 1.00000, 2.00000, 255, 0, 0, 1
```

Entrada: (teclado)

LZ

Saída: (tela)

```
1.00000, 1.00000, 1.00000, 255, 0, 0, 1
5.00000, 1.00000, 1.00000, 255, 0, 0, 1
10.00000, 1.00000, 1.00000, 255, 0, 0, 1
3.00000, 1.00000, 2.00000, 255, 0, 0, 1
8.00000, 1.00000, 2.00000, 255, 0, 0, 1
15.00000, 1.00000, 2.00000, 255, 0, 0, 1
```

Entrada: (teclado)

LI

Saída: (tela)

```
15.00000, 1.00000, 2.00000, 255, 0, 0, 1
8.00000, 1.00000, 2.00000, 255, 0, 0, 1
3.00000, 1.00000, 2.00000, 255, 0, 0, 1
1.00000, 1.00000, 1.00000, 255, 0, 0, 1
5.00000, 1.00000, 1.00000, 255, 0, 0, 1
10.00000, 1.00000, 1.00000, 255, 0, 0, 1
```

Entrada: (teclado)

LF

Saída: (tela)

```
10.00000, 1.00000, 1.00000, 255, 0, 0, 1
5.00000, 1.00000, 1.00000, 255, 0, 0, 1
1.00000, 1.00000, 1.00000, 255, 0, 0, 1
3.00000, 1.00000, 2.00000, 255, 0, 0, 1
8.00000, 1.00000, 2.00000, 255, 0, 0, 1
15.00000, 1.00000, 2.00000, 255, 0, 0, 1
```

Entrada: (teclado)

TOT

Saída: (tela)

LX:15 LI:6 LF:21

Entrada: (teclado)

PZ

Saída: (tela)

1.00000 1.00000 1.00000 TOTAL:1

Entrada: (teclado)

PI

Saída: (tela)

1.00000 1.00000 1.00000 TOTAL:6

Entrada: (teclado)

ID 1

Saída: (tela)

```
1.00000, 1.00000, 1.00000, 255, 0, 0, 1
3.00000, 1.00000, 2.00000, 255, 0, 0, 1
5.00000, 1.00000, 1.00000, 255, 0, 0, 1
8.00000, 1.00000, 2.00000, 255, 0, 0, 1
10.00000, 1.00000, 1.00000, 255, 0, 0, 1
15.00000, 1.00000, 2.00000, 255, 0, 0, 1
```

Entrada: (teclado)

CUT 0.0 0.0 0.0 10.0 10.0 10.0

Saída: (tela)

```
1.00000, 1.00000, 1.00000, 255, 0, 0, 1
3.00000, 1.00000, 2.00000, 255, 0, 0, 1
5.00000, 1.00000, 1.00000, 255, 0, 0, 1
8.00000, 1.00000, 2.00000, 255, 0, 0, 1
10.00000, 1.00000, 1.00000, 255, 0, 0, 1
```

=====

F.Osorio

09 Nov. 2021

=====