O TAD LEF

Este trabalho visa implementar um Tipo Abstrato de dado LEF - “lista de eventos futuros”, também conhecido como “fila de prioridade”.

Uma fila de prioridade é uma fila na qual cada item é associado a uma chave (prioridade); a fila é mantida ordenada segundo os valores crescentes das chaves. Simulações geralmente usam uma fila de prioridades chamada LEF (*lista de eventos futuros*), para manter os próximos eventos que irão ocorrer na simulação, na ordem crescente do tempo associado a cada evento. Nesse caso, a chave de cada evento é o seu instante de ocorrência.

Algumas operações tipicamente definidas sobre uma fila de prioridades são:

- `cria (fila, capacidade)` : cria uma nova fila, informando o número máximo de elementos que ela pode comportar; uma fila pode ser considerada infinita, se não tiver capacidade máxima definida.
- `destroi (fila)` : esvazia/destrói a fila
- `insere (fila, elemento, chave)` : insere um novo elemento na fila, posicionado de acordo com o valor da chave informada (prioridade)
- `elemento = retira_min (fila)` : retira o elemento da fila que tem a **menor** chave/prioridade
- `elemento = retira_max (fila)` : retira o elemento da fila que tem a **maior** chave/prioridade
- `valor_min (fila)` : informa o valor do elemento que tem a menor prioridade, sem alterar a fila
- `valor_max (fila)` : informa o valor do elemento que tem a maior prioridade, sem alterar a fila
- `chave_min (fila)` : informa o valor da menor prioridade, sem alterar a fila
- `chave_max (fila)` : informa o valor da maior prioridade, sem alterar a fila
- `tamanho (fila)` : informa o número atual de elementos na fila
- ...

Exemplo de funcionamento:

operação (valor, chave)	**conteúdo da fila**
cria	[ ]
insere (5, 3)	[**(5, 3)**]
insere (0, 6)	[(5, 3) **(0, 6)**]
insere (1, 1)	[**(1, 1)** (5, 3) (0, 6)]
insere (-2, 2)	[(1, 1) **(-2, 2)** (5, 3) (0, 6)]
insere (8, 2)[1)]	[(1, 1) (-2, 2) **(8, 2)** (5, 3) (0, 6)]
insere (9, 4)	[(1, 1) (-2, 2) (8, 2) (5, 3) **(9, 4)** (0, 6)]
retira min	[(-2, 2) (8, 2) (5, 3) (9, 4) (0, 6)]
retira min	[(8, 2) (5, 3) (9, 4) (0, 6)]
retira max	[(8, 2) (5, 3) (9, 4)]

Atividade

Você deve implementar um TAD LEF - “lista de eventos futuros” usando uma lista encadeada dupla. Essa lista deverá ser mantida em ordem crescente de acordo com o tempo (chave) de cada evento.

As seguintes operações devem ser suportadas:

- `c = cria (lef)`: cria uma nova LEF
- `destrói (lef)`: destrói a LEF
- `insere (lef, evento, tempo)`: insere o evento na LEF
- `retira (lef, evento, tempo)`: retira o evento da LEF
- `tamanho (lef)`: informa o número de eventos na LEF

- `imprime (lef)`: imprime o conteúdo da LEF

Os arquivos necessários para desenvolver este trabalho estão neste arquivo. Eles são:

- `lef.h`: arquivo de cabeçalho com os protótipos das funções (**não deve ser alterado**).
- `lef.c`: arquivo que implementa as operações sobre LEFs (“esqueleto” a completar).
- `teste.c`: código que testa a biblioteca (**não deve ser alterado**).
- `teste.txt`: saída do código que testa a biblioteca.
- `makefile`: arquivo do utilitário “make” para compilar seu código.

Entregáveis

Entregue um único arquivo `tp5.tgz` contendo todos os arquivos do projeto.

Critérios de avaliação:

- Funcionar corretamente
- Não ter problemas de memória (testar com Valgrind)
- Respeitar o conceito de TAD
- Usar funções para evitar repetição de blocos de código
- Respeitar estilo (endentação, espaçamento, nomes de variáveis)
- Usar comentários (nem demais, nem de menos)
- ... (outros que se fizerem necessários)

[1)]
Se dois valores têm a mesma chave, devem ser inseridos e retirados em ordem FIFO.

From:
https://wiki.inf.ufpr.br/maziero/ - **Prof. Carlos Maziero**

Permanent link:
https://wiki.inf.ufpr.br/maziero/doku.php?id=prog1:tad_lef

Last update: **2024/10/29 16:18**