

Relatório 1

Estudo de Estruturas de Dados e Seus Impactos na Execução de Algoritmos

Cristiano Lopes Moreira

Matrícula: 119103-0

Aluno		RA/Matrícula		Professor		Tipo	
Cristiano Lopes Moreira		119103-0		Dr Reinaldo Bianchi		Relatório de implementação	
Data	Versão	Turma		Nome do arquivo			Página
18/06/2019	2	1º. Semestre de 2019		PEL_216_Relatório_1_Cristiano_Moreira.doc			1 (13)

Sumário

1.	Introdução:	3
2.	Desenvolvimento teórico.	3
3.	Proposta de implementação.	7
4.	Experimentação e Resultados.	10
5.	Trabalhos Correlatos.	12
6.	Conclusão.	13

Aluno		RA/Matrícula		Professor		Tipo	
Cristiano Lopes Moreira		119103-0		Dr Reinaldo Bianchi		Relatório de implementação	
Data	Versão	Turma		Nome do arquivo			Página
18/06/2019	2	1º. Semestre de 2019		PEL_216_Relatório_1_Cristiano_Moreira.doc			2 (13)

1. Introdução

Uma das principais metodologias para averiguar o desempenho dos algoritmos é pela verificação da quantidade de tempo necessário para realizar suas operações e pela quantidade de acessos ao disco que são necessárias para realizar sua função. Dentre os pontos que impactam no desempenho dos algoritmos se destaca a estrutura de dados (sua forma de acesso, operações de consulta e de modificações de seu conteúdo dinâmico) uma vez que, por melhor que seja o algoritmo, seu tempo mínimo de execução será sempre, no mínimo, o tempo de acesso à sua estrutura de dados.

Estudar a estrutura de dados em Fila, na qual o conjunto de elementos inseridos primeiro são verificados pelas aplicações em primeiro lugar (FIFO – first in first out), e a estrutura de dados em Pilha, na qual o conjunto de elementos que são inseridos por último são verificados pelas aplicações em primeiro lugar (LIFO- last in first out) nos dá parâmetros para implementar algoritmos mais eficientes para solucionar os problemas de Busca em Largura e Busca em Profundidade, que utilizam essas técnicas de estrutura de dados em suas operações.

Neste trabalho iremos modelar os algoritmos de estrutura de dados Fila e Pilha, verificar sua operação de escrita e leitura, realizar sua análise assintótica no impacto de tempo para algoritmos que necessitam dessa forma de estrutura de dados, e quantificar o tempo dessas operações com o aumento de interações com a estrutura de dados.

2. Desenvolvimento teórico

As operações nas estruturas de dados são agrupadas em operações de consultas, que retornam a informação desejada, e operações de modificações, que possibilitam alterar o conjunto de informações contidas; basicamente, representadas por 5 operações primitivas:

Aluno		RA/Matrícula	Professor	Tipo	
Cristiano Lopes Moreira		119103-0	Dr Reinaldo Bianchi	Relatório de implementação	
Data	Versão	Turma	Nome do arquivo		Página
18/06/2019	2	1º. Semestre de 2019	PEL_216_Relatório_1_Cristiano_Moreira.doc		3 (13)

Busca (S,k): Verifica se o elemento 'k' está contido no agrupamento de dados 'S'.

Inserir (S,x): Insere o elemento 'x' no agrupamento de dados 'S'.

Remover (S,x): Remove o elemento 'x' no agrupamento de dados 'S'.

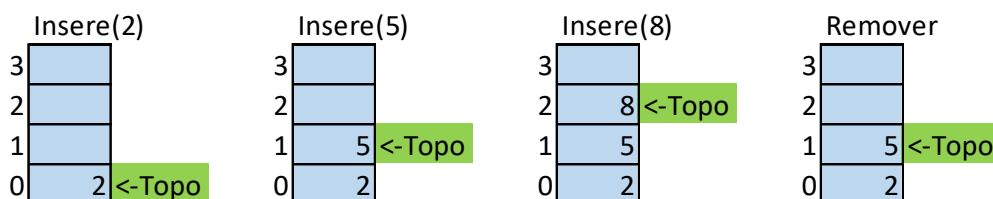
Minimo (S): Retorna o elemento de menor valor dentre os elementos do agrupamento 'S'.

Maximo (S): Retorna o elemento de maior valor dentre os elementos do agrupamento 'S'.

Pilhas e Filas são estruturas de dados dinâmicos que utilizam as operações "Inserir" e "Remover", como básicas, e algumas operações de suporte, como "Proximo" (topo), que informa qual a próxima informação a ser disponibilizada, e "Estado", que informa se está vazia ou cheia.

Pilhas

Em uma Pilha, o elemento eliminado do conjunto é o mais recentemente inserido, ela implementa a política LIFO (last in first out). Na estrutura de dados Pilha, é possível ter acesso a somente 01 item do agrupamento de dados por vez, o último a ser inserido. Essa estrutura de dados não se propõe a realizar busca de dados em sua cadeia. Para conhecer e utilizar o dado na penúltima posição de uma pilha é necessário remover o dado da última posição.



Pilhas se assemelham a um empilhamento de pratos, no qual o prato disponível e visível é o último prato que foi empilhado.

Aluno		RA/Matrícula		Professor	Tipo
Cristiano Lopes Moreira		119103-0		Dr Reinaldo Bianchi	Relatório de implementação
Data	Versão	Turma		Nome do arquivo	Página
18/06/2019	2	1º. Semestre de 2019		PEL_216_Relatório_1_Cristiano_Moreira.doc	4 (13)

As operações utilizadas:

- Iniciar Pilha
- Inserir na Pilha (empilhar)
- Remover da Pilha (desempilhar)
- Topo (Mostrar informação no topo da pilha)
- Estado, verificar estado da Pilha (pilha cheia, vazia, normal)

Filas

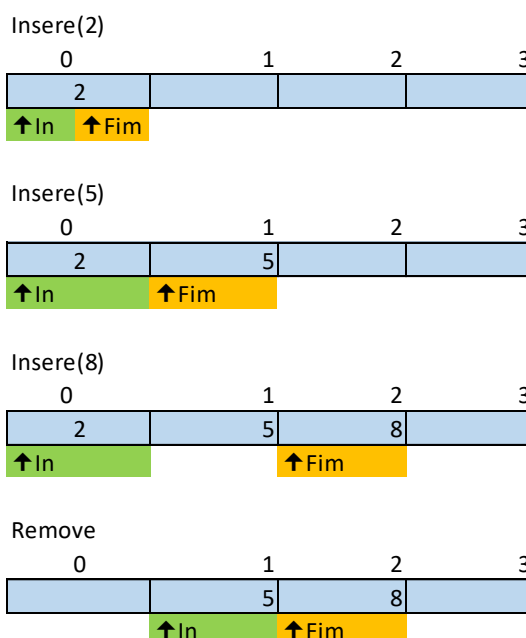
Em uma Fila, o elemento eliminado do conjunto é o mais antigo inserido, é implementada a política FIFO (first in first out). Na estrutura de dados Fila, é possível ter acesso a somente a 01 item do agrupamento de dados por vez, o primeiro a ser inserido. Essa estrutura de dados, assim como a Pilha, não se propõe a realizar busca de dados em sua cadeia. Para conhecer e utilizar o dado na segunda posição de uma pilha é necessário remover o dado da primeira posição.

Filas se assemelham a uma fileira de pessoas em uma caixa registradora, a primeira pessoa a ser atendida será a primeira pessoa que chegou na fila, e a segunda a ser atendida será a segunda que chegou na fila e assim por diante.

Uma característica da Fila, distinta da Pilha, é que a fila trabalha com dois apontadores, o início da Fila (cabeça) e o fim da Fila (cauda); quando um elemento é inserido na Fila ele ocupa seu lugar no final Fila. O elemento retirado da Fila é sempre o que está no início da fila.

Aluno		RA/Matrícula	Professor	Tipo	
Cristiano Lopes Moreira		119103-0	Dr Reinaldo Bianchi	Relatório de implementação	
Data	Versão	Turma	Nome do arquivo		Página
18/06/2019	2	1º. Semestre de 2019	PEL_216_Relatório_1_Cristiano_Moreira.doc		5 (13)

Pelo uso dos apontadores, que acompanham o início e fim da fila, o algoritmo se norteia na inserção e remoção de um elemento da Fila. Quando entra um novo elemento, o apontador Fim é acrescido de +1, quando é removido um elemento o apontador Início é acrescido de +1. Esta rotina dinâmica de apontadores possibilita o uso circular da memória, já que toda vez que é liberado um elemento da Fila o “espaço” que este elemento ocupava se torna disponível. Para a utilização circular da memória é necessário implementar uma rotina que verifica se os apontadores alcançaram o final da memória. Quando isso ocorre, verifica-se se o início da memória está disponível, não marcada por nenhum apontador. Caso esteja disponível, é realizada uma volta reiniciando o apontador para primeira posição da memória.



As operações utilizadas:

- Iniciar Fila
- Inserir na Fila (enfileirar)
- Remover da Fila (desenfileirar)
- Primeiro (Mostrar informação do primeiro elemento da Fila)

Aluno		RA/Matrícula		Professor		Tipo	
Cristiano Lopes Moreira		119103-0		Dr Reinaldo Bianchi		Relatório de implementação	
Data	Versão	Turma		Nome do arquivo			Página
18/06/2019	2	1º. Semestre de 2019		PEL_216_Relatório_1_Cristiano_Moreira.doc			6 (13)

Para a avaliação do desempenho de uma estrutura de dados, em geral, verifica-se o tempo de execução com a aplicação a que se destina levando-se em consideração o tamanho do conjunto de dados.

3. Proposta de implementação

É proposta a implementação de 2 objetos, um para a utilização da estrutura de dados em Pilha e outro para a estrutura de dados em Fila; ambos os algoritmos devem ser avaliados pelo tempo de execução de seus principais métodos, inserir e remover, na manipulação da estrutura de dados de 0 a 100.000.000 de manipulação de dados, com granularidade de 1.000.000 interações.

Estrutura de dados Pilha

O Algoritmo da estrutura de dados Pilha será implementado via Classe, com métodos para Inserir, responsável por enviar a informação que se deseja inserir na estrutura de dados; Remover, responsável por retirar a informação armazenada no topo da pilha; Estado, responsável por informar se a estrutura de dados está vazia, quando não existem dados armazenados na Pilha, cheia, quando a pilha está utilizando a capacidade máxima de memória reservada para a estrutura de dados, e normal, quando a estrutura de dados já contém dados e está pronta para receber mais informações; e um método adicional, PegaTopo, responsável por mostrar a informação contida no topo da estrutura de dados.

O Objeto Pilha

Pilha
- ponteiroPilha: interger - capacidade: interger - index: interger
<<constructor>> Pilha(memInicial: int) + PegaTopo() + AdicionaPilha(Valor: int) + RemovePilha() + Estado() + dimensiona(Memoria: int)

Aluno		RA/Matrícula		Professor		Tipo	
Cristiano Lopes Moreira		119103-0		Dr Reinaldo Bianchi		Relatório de implementação	
Data	Versão	Turma		Nome do arquivo			Página
18/06/2019	2	1º. Semestre de 2019		PEL_216_Relatório_1_Cristiano_Moreira.doc			7 (13)

Pseudocódigo:

Estado(S)

Se S.topo==0

Retorna vazio

Se não

Se S.topo==Capacidade
retorna cheio.

Se nao

retorna normal.

Inserir(S,x)

S.topo= S.topo+1

S.topo[S.topo]= x

Remover(S)

Se Estado(S)==vazio

Retorna “não existem itens para excluir”

Se não

S.topo= S.topo -1

Retorna S.topo[S.topo+1]

Estrutura de dados Fila

O Algoritmo da estrutura de dados Fila será implementado via Classe, com métodos para Inserir, responsável por enviar a informação que se deseja inserir na estrutura de dados; Remover, responsável por retirar a informação armazenada no início da Fila; e Estado, responsável por informar se a estrutura de dados está vazia, quando não existem dados armazenados na Fila, cheia, quando a Fila está utilizando a capacidade máxima de memória reservada para a estrutura de dados, e normal, quando a estrutura de dados já contém dados e está pronta para receber mais informações. Na estrutura de dados Fila serão utilizadas duas variáveis privadas como ponteiros, início e fim, que irão apontar em uma matriz de memória a posição de início e fim da Fila. Como artifício será também será utilizada uma variável de quantidade de elementos na Fila. Um método adicional, PegaSaida, responsável por mostrar a informação contida na posição inicial da estrutura de dados.

Aluno		RA/Matrícula	Professor	Tipo	
Cristiano Lopes Moreira		119103-0	Dr Reinaldo Bianchi	Relatório de implementação	
Data	Versão	Turma	Nome do arquivo		Página
18/06/2019	2	1º. Semestre de 2019	PEL_216_Relatório_1_Cristiano_Moreira.doc		8 (13)

O Objeto Fila

Fila
- ponteiroFila: interger
- idXAdd: interger
- idxRem: interger
- capacidade: interger
- index: interger
<<constructor>> Pilha(memInicial: int)
+ PegaSaida()
+ AdicionaFila(Valor: int)
+ RemoveFila()
+ Estado()
+ dimensiona(Memoria: int)

Pseudocódigo:

Estado(S)

```

Se S.elementos==0
    Retorna vazio
Se não
    Se S.elementos==Capacidade
        retorna cheio.
    Se nao
        retorna normal.

```

Inserir(S,x)

```

Se Estado(S)==Cheio
    Retorna “não espaço para inserir novos dados”
Se não
    S.[S.idxFim]=x
    S.elementos= S.elementos+1
    Se S.idxFim ==Capacidade
        S.idxFim=0
    Se não
        S.idxFim=S.idxFim+1

```

Remover(S)

```

Se Estado(S)==vazio
    Retorna “não existem itens para excluir”
Se não
    tmp= S[S.idxInicio].valor

```

Aluno		RA/Matrícula	Professor	Tipo	
Cristiano Lopes Moreira		119103-0	Dr Reinaldo Bianchi	Relatório de implementação	
Data	Versão	Turma	Nome do arquivo		Página
18/06/2019	2	1º. Semestre de 2019	PEL_216_Relatório_1_Cristiano_Moreira.doc		9 (13)

```

Se S.idxInicio == Capacidade
    S.idxInicio = 0
Se não
    S.idxInicio = S.idxInicio + 1
Retorna tmp
    
```

As estruturas de dados Fila e Pilha não são estruturas eficientes para localizar informações dentro da base de dados e também não são eficientes nas operações de mínimo e máximo, já que não se propõem a ordenar os dados pelo seu valor e sim pela sua posição de entrada na estrutura.

4. Experimentação e Resultados

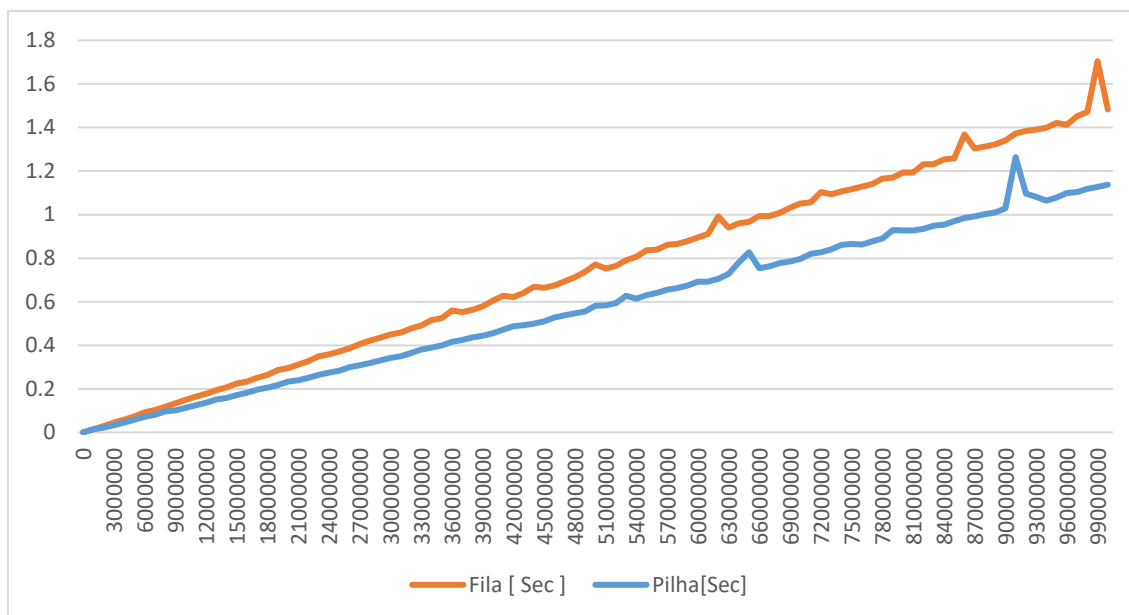
A interação com a estrutura de dados de Lista ou Fila tem tempo de execução na interação com o algoritmo, para cada interação o tempo de resposta linear $O(1)$, e seu consumo de tempo para n interações estará em $O(n)$, visto nos resultados abaixo.

n interações	Fila [Sec]	Pilha[Sec]
0	0	0
1000000	0.015	0.014
2000000	0.029	0.022
3000000	0.045	0.034
4000000	0.058	0.045
5000000	0.073	0.059
6000000	0.092	0.071
7000000	0.103	0.08
8000000	0.117	0.096
9000000	0.133	0.101
10000000	0.15	0.113
11000000	0.165	0.125
12000000	0.177	0.136
13000000	0.194	0.151
14000000	0.207	0.159
15000000	0.224	0.172
16000000	0.234	0.182
17000000	0.251	0.196

Aluno		RA/Matrícula	Professor	Tipo	
Cristiano Lopes Moreira		119103-0	Dr Reinaldo Bianchi	Relatório de implementação	
Data	Versão	Turma	Nome do arquivo		Página
18/06/2019	2	1º. Semestre de 2019	PEL_216_Relatório_1_Cristiano_Moreira.doc		10 (13)

18000000	0.264	0.205
19000000	0.286	0.217
20000000	0.295	0.234
21000000	0.311	0.239
22000000	0.328	0.251
23000000	0.349	0.264
24000000	0.358	0.274
25000000	0.372	0.284
26000000	0.387	0.299
27000000	0.406	0.309
28000000	0.421	0.318
29000000	0.435	0.33
30000000	0.45	0.342
31000000	0.459	0.349
32000000	0.478	0.364
33000000	0.49	0.38
34000000	0.515	0.389
35000000	0.525	0.399
36000000	0.56	0.415
37000000	0.553	0.424
38000000	0.563	0.436
39000000	0.579	0.444
40000000	0.605	0.456
41000000	0.628	0.472
42000000	0.621	0.487
43000000	0.641	0.492
44000000	0.668	0.5
45000000	0.664	0.51
46000000	0.675	0.528
47000000	0.694	0.537
48000000	0.712	0.547
49000000	0.738	0.556
50000000	0.771	0.582

Aluno		RA/Matrícula		Professor		Tipo	
Cristiano Lopes Moreira		119103-0		Dr Reinaldo Bianchi		Relatório de implementação	
Data	Versão	Turma		Nome do arquivo		Página	
18/06/2019	2	1º. Semestre de 2019		PEL_216_Relatório_1_Cristiano_Moreira.doc		11 (13)	



As interações com os ponteiros utilizados pela estrutura de dados da Fila, comparada à estrutura de dados Pilha, necessita mais tempo de execução por que tanto para a inserção quanto para a remoção, dos dados da estrutura de dados, é necessário realizar interações com os ponteiros que indicam o início e o fim da fila. Ambas as estruturas de dados têm o tempo de execução linear.

5. Trabalhos Correlatos

VUILLEMIN, J. Manipulating Priority Queues. Communications of the ACM, Volume 21, Number 4, p.309-315, April 1978

JONES, D. W. An Empirical Comparison of Priority-Queue and Event-Set Implementaitons. Communications of the ACM, Volume 29, Number 4, p.300-311, April 1986

Aluno		RA/Matrícula	Professor	Tipo
Cristiano Lopes Moreira		119103-0	Dr Reinaldo Bianchi	Relatório de implementação
Data	Versão	Turma	Nome do arquivo	Página
18/06/2019	2	1º. Semestre de 2019	PEL_216_Relatório_1_Cristiano_Moreira.doc	12 (13)

6. Conclusão

Os resultados mostraram que o tempo de execução de um algoritmo está diretamente vinculado à estrutura de dados utilizada e à forma de interação entre o algoritmo e essa estrutura de dados, o que é verificado entre a diferença de interação com ponteiros da Fila e Pilha.

Os algoritmos que utilizam estrutura de dados Fila e Pilha, como busca em profundidade e busca em largura, em sua interação com múltiplos elementos terão um tempo de execução no mínimo linear, já que apenas para realizar as interações com a estrutura de dados é necessário um tempo de execução crescente linear, o que nos mostra que tais algoritmos não podem ter tempo de execução inferior à $O(n)$.

Assim concluímos que, para implementar um algoritmo melhor e mais eficiente, que solucione um determinado problema, faz-se necessário conhecer o formato e a linguagem que descrevem o problema e encontrar uma estrutura de dados que melhor se adeque a esse problema. Para isso é necessário conhecer o tempo de execução e o formato de manipulação das diversas estruturas de dados, de forma a possibilitar escolher aquela que mais se adeque à solução apresentada para um dado problema.

Aluno		RA/Matrícula	Professor	Tipo	
Cristiano Lopes Moreira		119103-0	Dr Reinaldo Bianchi	Relatório de implementação	
Data	Versão	Turma	Nome do arquivo		Página
18/06/2019	2	1º. Semestre de 2019	PEL_216_Relatório_1_Cristiano_Moreira.doc		13 (13)