Notas de aula 11/04

Exemplos de aplicações de pilhas e filas

Notação polonesa inversa:

Na notação tradicional de expressões matemáticas (notação infixa) o operador sempre aparece entre os seus dois operandos:

a + b

a - b * c

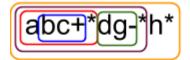
Já na notação pós-fixa, também conhecida como notação polonesa inversa, o operador é colocado após os seus dois operandos. Exemplos:

Notação tradicional	Notação polonesa inversa
a-b	ab-
a-b*c	abc*-
(a-b)*c	ab-c*
a*(b+c)*(d-g)*h	abc+*dg-*h*

Neste tipo de notação os parênteses são dispensados. A ordem das operações é dada pela ordem em que os operadores aparecem. Por exemplo:



- O primeiro operador que aparece é o *, então multiplica-se b e c.
- O segundo operador que aparece é o -, então multiplica-se a e o resultado da op. anterior.



- O primeiro operador que aparece é o +, então soma-se b e c.
- O segundo operador que aparece é o *, então multiplica-se a e o resultado da op. anterior.
- O terceiro operador que aparece é o -, é feita a op. d - g.
- O quarto operador que aparece é o *, então multiplica-se o resultado de abc+* e dg-
- O último operador é o *, então o resultado anterior e multiplicado por h.

A notação polonesa ganhou destaque na década de 70, por sua adoção nas calculadoras HP.

Esse tipo de notação pode ser facilmente resolvido usando uma estrutura do tipo **pilha**. Considere a expressão:

que na notação polonesa inversa fica:

Percorrendo a expressão da esquerda para a direita: cada vez que aparece um número, empilhamos. Cada vez que aparece um operador, desempilhamos os dois últimos números, calculamos o resultado da operação e empilhamos o resultado:

Próximo	Operações necessárias	Pilha
5	empilha(5)	5
1	empilha(1)	5, 1
2	empilha(2)	5, 1, 2
+	desempilha(2) desempilha(1) soma(1, 2) empilha(3)	5, 3
4	empilha(4)	5, 3, 4
*	desempilha(4) desempilha(3) multiplica(3, 4) empilha(12)	5, 12
+	desempilha(12) desempilha(5) soma(5, 12) empilha(17)	17
3	empilha(3)	17, 3
-	desempilha(3) desempilha(17) subtrai(17, 3) empilha(14)	14

O item final da pilha é o resultado da expressão.

Validação de expressões com parênteses, colchetes e chaves

Expressões matemáticas frequentemente contém parênteses (), colchetes [] e chaves {}. Para a expressão estar bem formada, todos os símbolos abertos tem que ser fechados.

Exemplos de combinações bem formadas:

```
(()[()])
((([])())[])
({()}[])
```

Exemplos de combinações mal formadas:

```
([)]
((([])())[)
{()([]))
```

Esse tipo de problema também pode ser facilmente resolvido com auxílio de uma estrutura do tipo **pilha**. A idéia é simples:

Ex.:((([])())[])

Próximo	Operações necessárias	Pilha
(empilha ((
(empilha (((
(empilha ((((
[empilha [])))
1	se o topo da pilha contém [desempilha [se não para (a expressão é mal formada)	(((
)	se o topo da pilha contém (desempilha (se não para (a expressão é mal formada)	((
(empilha ((((
)	se o topo da pilha contém (desempilha (se não para (a expressão é mal formada)	((
)	se o topo da pilha contém (desempilha (se não para (a expressão é mal formada)	(
[empilha [([
]	se o topo da pilha contém [desempilha [se não para (a expressão é mal formada)	(
)	se o topo da pilha contém (desempilha (se não	

|--|

A pilha resultante está vazia, isso quer dizer que para cada símbolo aberto, havia um símbolo fechado na posição esperada. Logo, a expressão está bem formada.

Ex.: ([)]

Próximo	Operações necessárias	Pilha
(empilha ((
[empilha [])
)	se o topo da pilha contém (desempilha (se não para (a expressão é mal formada)])

A pilha resultante não está vazia, então a expressão está mal formada.

Flla prioritária

Considere um banco que utiliza duas filas de atendimento: uma fila normal e uma fila para as pessoas com prioridade (idosos e gestantes, por exemplo). Sempre que um guichê de atendimento é liberado, se houver alguém na fila prioritária, é chamado um cliente dessa fila. Se não houver ninguém na fila prioritária, é chamado um cliente da fila normal.

Abaixo um exemplo de sequência de eventos que pode acontecer no banco. Considere que N indica a chegada de um cliente que entra na fila normal, I ou G indica a chegada de um cliente que entra na fila preferencial, C indica que um guichê foi liberado e uma pessoa será atendida e F indica que o banco fechou e ninguém mais vai ser atendido. O número ao lado de um N, I, ou G, indica o código do cliente.

N 6

N 7

N 8

۱9

I 85

G 23

С

С

С

C F

Como podemos criar um algoritmo que controla a ordem em que as pessoas serão chamadas? Podemos pensar em duas filas N (normal) e P (preferencial):

Próximo	Operações necessárias	Flias
N 6	enfileira(N, 6)	N: 6 P:
N 7	enfileira(N, 7)	N: 6, 7 P:
N 8	enfileira (N, 8)	N: 6, 7, 8 P:
19	enfileira(P, 9)	N: 6, 7, 8 P: 9
I 85	enfileira(P, 85)	N: 6, 7, 8 P: 9, 85
G 23	enfileira(G, 23)	N: 6, 7, 8 P: 9, 85, 23
С	se P não está vazia: desenfileira(P) se não: desenfileira(N)	N: 6, 7, 8 P: 85, 23
С	se P não está vazia: desenfileira(P) se não: desenfileira(N)	N: 6, 7, 8 P: 23
С	se P não está vazia: desenfileira(P) se não: desenfileira(N)	N: 6, 7, 8 P:
С	se P não está vazia: desenfileira(P) se não: desenfileira(N)	N: 7, 8 P:
F	para	N: 7, 8 P: