Pilhas

Aula 03

Ivone P. Matsuno Yugoshi
ivone.matsuno@ufms.br

Ronaldo Fiorilo dos Santos
ronaldo.santos@ufms.br

Universidade Federal de Mato Grosso do Sul
Câmpus de Três Lagoas
Bacharelado em Sistemas de Informação

Algoritmos e Programação II

Slides baseados no material do Prof. Fábio Henrique Viduani Martinez - FACOM/UFMS

Definição

- Uma **pilha** é uma lista linear tal que as operações de inserção e remoção são realizadas em um único extremo dessa estrutura de dados.
  - baseada no princípio LIFO (*Last In, First Out*)
- O funcionamento dessa lista linear pode ser comparado a qualquer pilha de objetos que usamos com frequência como, por exemplo, uma pilha de pratos de um restaurante. Em geral, os clientes do restaurante retiram pratos do início da pilha, isto é, do primeiro prato mais alto na pilha.
- Os funcionários colocam pratos limpos também nesse mesmo ponto da pilha. Seria estranho ter de movimentar pratos, por exemplo no meio ou no final da pilha, sempre que uma dessas dessas operações fosse realizada.

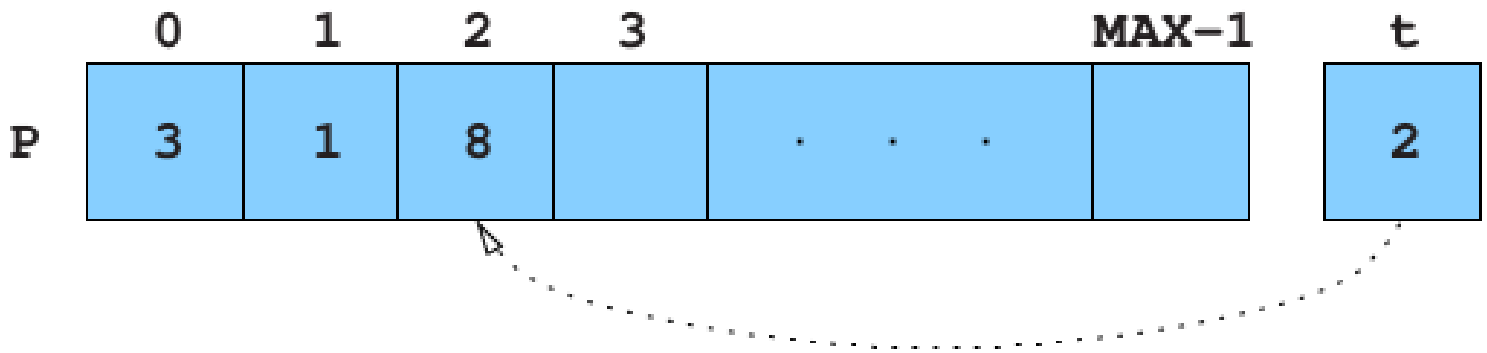
Definição

- O indicador do extremo onde ocorrem as operações de inserção e remoção é chamado de **topo** da pilha. Essas duas operações são também chamadas de **empilhamento** e **desempilhamento** de elementos.
- Nenhuma outra operação é realizada sobre uma pilha, a não ser em casos específicos.
- Observe, por exemplo, que a operação de busca não foi mencionada e não faz parte do conjunto de operações básicas de uma pilha.

Operações básicas em alocação sequencial

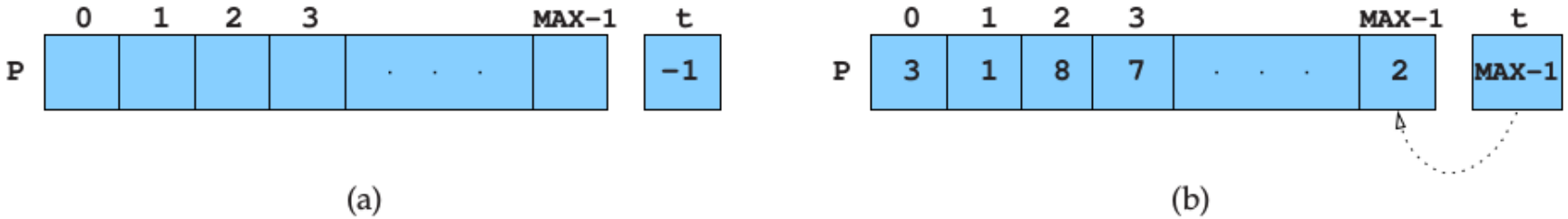
- Uma pilha em alocação sequencial é descrita através de duas variáveis: um vetor e um índice para o vetor.
- Supomos que a pilha esteja armazenada no vetor `P[0..MAX-1]` e que a parte do vetor efetivamente ocupada pela pilha seja `P[0..t]`, onde `t` é o índice que define o topo da pilha.
- Os compartimentos desse vetor são convencionalmente do tipo `int`, mas podemos defini-los de qualquer tipo.

Operações básicas em alocação sequencial



Operações básicas em alocação sequencial

- Convencionamos que uma pilha está vazia se seu topo `t` vale `-1` e cheia se seu topo `t` vale `MAX-1`.



- A declaração de uma pilha e sua inicialização são mostradas abaixo:

```
int t, P[MAX];
t = -1;
```

Operações básicas em alocação sequencial

- A operação de inserir um objeto em uma pilha, ou **empilhar**, é descrita na função `empilha_seq` a seguir.

```
/* Recebe um ponteiro t para o topo de uma pilha P
e um elemento y e insere y no topo da pilha P */
void empilha_seq(int *t, int P[MAX], int y)
{
    if (*t != MAX - 1) {
        (*t)++;
        P[*t] = y;
    }
    else
        printf("Pilha cheia!\n");
}
```

Operações básicas em alocação sequencial

- A operação de remover, ou **desempilhar**, um objeto de uma pilha é descrita na função `desempilha_seq` a seguir.

```
/* Recebe um ponteiro t para o topo de uma pilha
P e remove um elemento do topo da pilha P */
int desempilha_seq(int *t, int P[MAX])
{
    int r;
    if (*t != -1) {
        r = P[*t];
        (*t)--;
    } else {
        r = INT_MIN;
        printf("Pilha vazia!\n");
    }
    return r;
}
```

Exercícios

1. Uma palavra é um **palíndromo** se a sequência de caracteres que a constitui é a mesma quer seja lida da esquerda para a direita ou da direita para a esquerda. Por exemplo, as palavras RADAR e MIRIM são palíndromos. Escreva um programa eficiente para reconhecer se uma dada palavra é palíndromo.

2. Considere o problema de decidir se uma dada sequência de parênteses e chaves é bem-formada, ou seja, parênteses e colchetes são fechados na ordem inversa àquela em que foram abertos. Por exemplo, a sequência: (( ) [( )]) é bem-formada,
enquanto que a sequência: ([ )] é malformada.
Suponha que a sequência de parênteses e chaves está armazenada em uma cadeia de caracteres. Escreva uma função bemformada que receba a cadeia de caracteres s e devolva 1 se s contém uma sequência bem-formada de parênteses e colchetes e devolva 0 se a sequência está malformada.