Módulo 6

Pilhas

Algoritmos e Estruturas de Dados II

C++

(Rone Ilídio)

Pilha

- Sequência de elementos na memória
- Implementação
  - Vetor
  - Lista
- Funções
  - Inserção: sempre no topo
  - Remoção: o último a chegar, primeiro a sair
  - Imprime
- Política LIFO: Last In First Out

Pilha - Vetor

- Vetor de 0 a n-1
- Variável:
  - Topo da pilha: t 0 =< t < n
- Pilha cheia
  - t = n
- Fila vazia
  - t = 0

Pilha – Estrutura de Dados

```
struct cel{
   int conteudo;
};
typedef cel celula;

int const n = 4;

celula f[n];
int t=0;
```

Inserir

- Verifica se a pilha está cheia e insere

```
void insere(int y){
    if(t<n){
       f[t].conteudo = y;
       t++;
    }
    else cout << "\nPilha cheia";
}
```

Pilha - Remove

```
celula remove(){
    celula x;
    x.conteudo = -1; //valor nulo
    if (t>0){
        t--;
        x = f[t];
    }
    else {
        cout << "\nPilha Vazia!";
    }
    return x;
}
```

Pilha - Imprime

```
void imprime(){
    cout << "\nPilha:(t="<<t<<"): ";
     int a = t-1;
     while(a>=0){
         cout<<" "<<f[a].conteudo;
         a--;
     }
}
```

Exemplo Main

```
int main(){
    insere(5);insere(3);
    insere(2);insere(4);
    imprime();
    celula r = remove();
    imprime();
    insere(7);
    imprime();
    insere(9);
    imprime();
    getch();
}
```

Pilha – Lista Encadeada

Pilha – Lista Encadeada

```
int main(){
    celula cabeca;
    celula *p;
    p = &cabeca;
    p->seg = NULL;

    Empilha(5,p);Empilha(3,p);
    Empilha(2,p);Empilha(4,p);
    Empilha(8,p);
    int r = Desempilha(p);
    cout << "\nDesempilhou " << r;
    imprime(p);
 getch();
}
```

Pilha – Lista Encadeada

- Insere ou Empilha

```
void Empilha(int y, celula *p){
    celula *nova = new celula;
    nova->conteudo = y;
    nova->seg = p->seg;
    p->seg = nova;
}
```

Pilha – Lista Encadeada

- Remove ou Desempilha

```
int Desempilha(celula *p){
    int x;
    celula *q;
    q = p->seg;
    x = q->conteudo;
    p->seg = q->seg;
    free(q);
    return x;
}
```