



Anhanguera

*Aqui o seu esforço
ganha força.*



Anhanguera

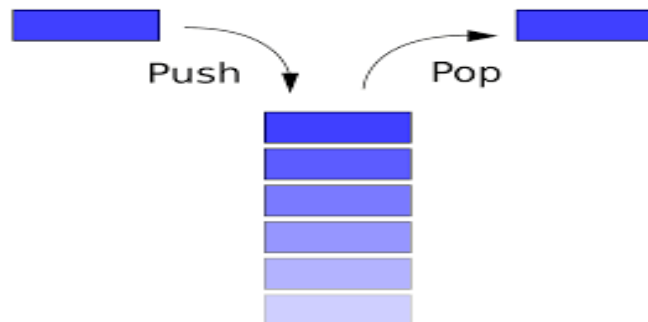
Pilhas com alocação dinâmica

Prof. Esp. Rodrigo Hentz



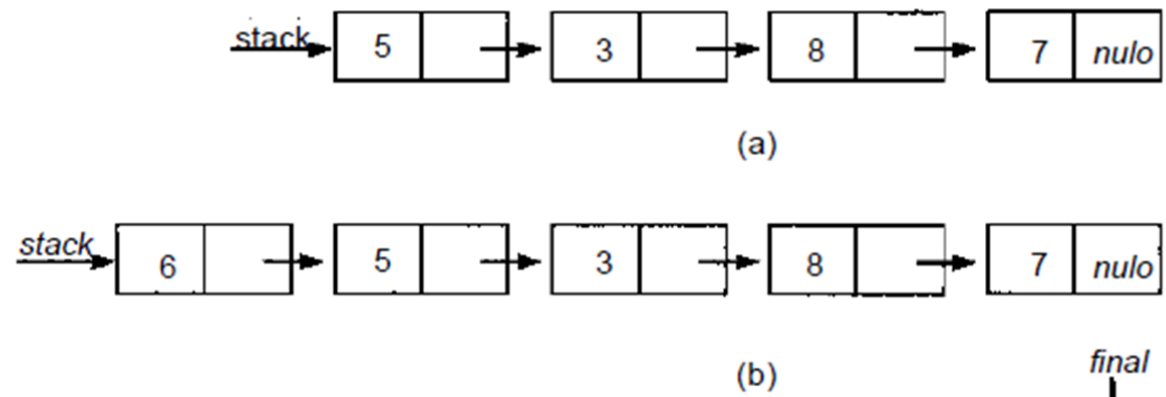
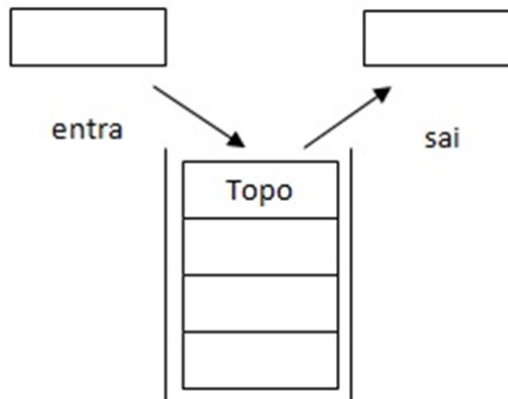
Pilhas

- A pilha é um objeto dinâmico e mutável que possui um conjunto ordenado de itens onde novos itens podem ser inseridos e excluídos apenas de seu “topo”.
- Implementa o conceito de LIFO (last-in, first-out) ou UEPS (último a entrar é o primeiro a sair).
- Podem ser implementados com vetores ou alocação dinâmica



Pilhas

- Operações principais:
- Empilhamento (push)
- Desempilhamento (pop)



Pilhas

Outras operações:

- » empty(pilha)
 - Retorna se a pilha está vazia
- » stacktop(pilha)
 - Retorna o item superior da pilha
- » inicialize(pilha)
 - Inicializa uma pilha vazia

Pilhas

- Na implementação de pilhas com alocação dinâmica não temos limite de espaço (desde que a alocação seja realizada com sucesso)
- Desta maneira não implementamos a operação full(pilha) para verificar se a mesma está cheia

```
typedef struct no {  
    int info;  
    struct no* next;  
} sPilha, sNo;  
  
sNo* criarNo(int valor)  
{  
    sNo* p = (sNo*)malloc(sizeof(sNo));  
    p->info = valor;  
    p->next = NULL;  
    return p;  
}
```

```
sPilha* initialize()
{
    printf("\nPilha inicializada\n");
    return NULL;
}

int empty(sPilha* pilha)
{
    return pilha == NULL;
}

sPilha* push(sPilha* pilha, int valor) //inclui, empilha
{
    sNo* p = criarNo(valor);
    p->next = pilha;
    return p;
}
```



```
int pop(sPilha** pilha) //retira, desempilha
{
    int num; sNo* p;
    if (empty(*pilha))
    {
        printf ("\nPilha esta vazia.");
        num = *pilha;
    }
    else
    {
        p = (*pilha);
        num = p->info;
        *pilha = p->next;
        free(p);
    }
    return num;
}
```

```
int stacktop(sPilha* pilha)
{
    int num;
    if (empty(pilha))
    {
        .....
        printf ("\nPilha esta vazia.");
        num = pilha;
    }
    else
    {
        .....
        num = pilha->info;
    }
    return num;
}
```

```
void print(sPilha* pilha)
{
    sNo* p;
    if (empty(pilha)) printf ("\nPilha esta vazia.");
    else
        for (p = pilha; p != NULL; p = p->next)
            printf("\nInfo = %d", p->info);
}
```

```
int main(int argc, char *argv[]) {
    sPilha* pilha; int opcao, num;
    do
    {
        printf("\n");
        printf("1 - Inicializa\n");
        printf("2 - POP\n");
        printf("3 - PUSH\n");
        printf("4 - STACKTOP\n");
        printf("5 - PRINT\n");
        printf("0 - SAIR\n");
        printf("\nEntre com a opcao: "); scanf("%d", &opcao);
        switch (opcao)
        {
            case 1:
                pilha = initialize();
                break;
            case 2:
                printf("\nNumero desempilhado %d", pop(&pilha)); printf("\n");
                break;
```

```
case 3:
    printf ("\nEntre com o numero para empilhar: "); scanf ("%d", &num); printf("\n");
    pilha = push(pilha, num);
    break;
case 4:
    printf("\nNumero no topo %d", stacktop(pilha)); printf("\n");
    break;
case 5:
    print(pilha); printf("\n");
    break;
}
fflush(stdin);
} while (opcao != 0);
return 0;
}
```



Anhanguera

*Aqui o seu esforço
ganha força.*