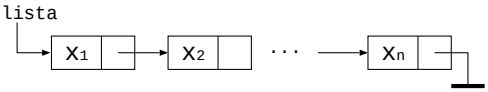
Listas Encadeadas com Cabeça e Listas Circulares

Listas Encadeadas

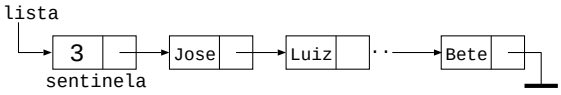
Problemas:

- todo percurso requer teste de término da lista
- inserção ou eliminação no início da lista altera o conteúdo do ponteiro `l` (aponta para o primeiro nó da lista)



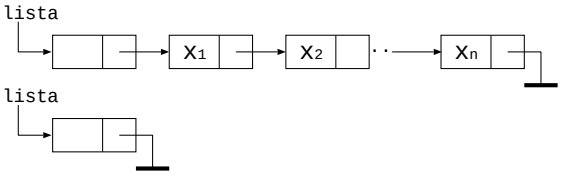
Exemplo de lista com cabeça

Armazenar número de elementos da lista, para que não seja necessário atravessá-la contando seus elementos:



Nó Cabeça

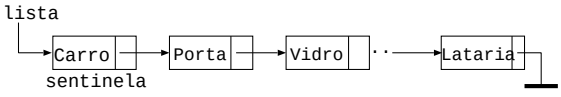
- nó especial (*head node*) sem informação
- aponta para o nó que contém x_1
- pode guardar outras informações pertinentes
- simplifica algumas operações



O conteúdo do primeiro nó é irrelevante: serve apenas para marcar o início da lista. O primeiro nó é a cabeça (= head node = head cell = dummy cell) da lista e está sempre no mesmo lugar na memória, mesmo que a lista fique vazia.

Exemplo de lista com cabeça

Em uma fábrica, guarda-se as peças que compõem cada equipamento produzido, sendo este indicado pelo nó sentinela:



Remove em lista sem cabeça

```
int remove_inicio(Lista *p_l, elem_t *p_e){
  No_lista *aux;
  if (!vazia(p_l)){
    aux = *p_l;
    *p_e = aux->info;
    if (aux->prox == NULL)
      *p_l = NULL; // retirou o unico elem
    else
      *p_l = aux->prox;
    free(aux);
    return 1;
  }
  return 0;
}
```

Listas Circulares

Listas encadeadas:

- não é possível atingir o nó anterior
- o ponteiro externo deve ser preservado

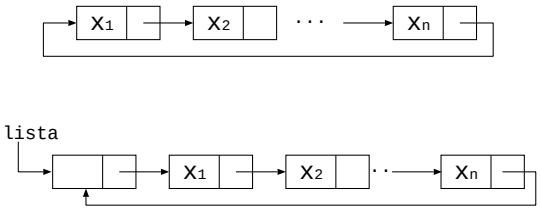
Listas circulares:

- último nó aponta para o primeiro
- com ou sem nó cabeça

Remove em lista com cabeça

```
int remove_inicio(Lista *p_l, elem_t *p_e){
    No_lista *aux;
    aux = p_l->prox;
    if (!vazia(p_l)) {
        p_l->prox = aux->prox; // p_l->prox->prox
        *p_e = aux->info;
        free (aux);
        return 1;
    }
    return 0;
```

Lista Circular



Nó cabeça

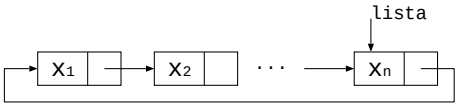
- pode guardar informações como: tamanho, ponteiro para o nó atual, ponteiro para o último nó, ordenada ou não, ...
- é possível implementar o nó cabeça com uma estrutura diferente dos outros nós

```
struct cab{
  int tam;
  No_lista *fim;
  No_lista *ini;
};
```

Lista Circular

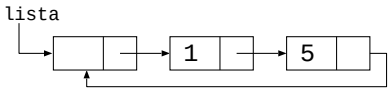
lista aponta para o último nó:

- acesso ao último nó
- inserir e eliminar um elemento a partir do início ou do final da lista



Como representar números grandes?

Utilizando listas: 15



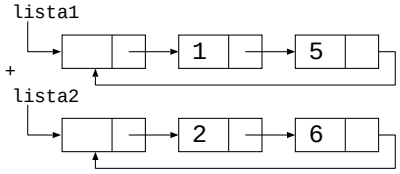
Implementação

- números ao contrário:
  num MOD 10 e insere no final da lista
  num DIV 10
- como colocar o número somado em uma nova lista?
  soma MOD 10 é inserido no início da lista
- como recuperar o sobe 1?
  soma DIV 10

Soma

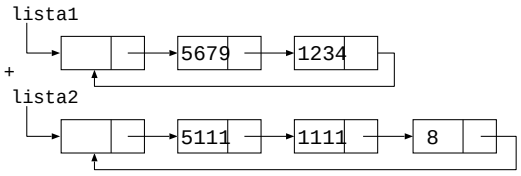
Para somar dois números representados em listas, somam-se blocos de memória dois a dois, da direita para a esquerda

Exemplo: 15+26



Otimização de memória

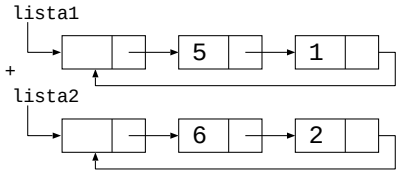
12.345.679 + 811.115.111



Facilitando

Para facilitar, os números podem ser representados ao contrário

Exemplo: 15+26



Exercício

Implemente a soma de dois números grandes com a otimização de memória.

Bibliografia

- Zhao Liang, material de aula da disciplina *Algoritmos e Estruturas de Dados I* ICMC - USP - São Carlos
- Thiago A. S. Pardo, material de aula da disciplina *Algoritmos e Estruturas de Dados I* ICMC - USP - São Carlos