Listas lineares duplamente encadeadas

Aula 22

Fábio Henrique Viduani Martinez Diego Padilha Rubert

Faculdade de Computação
Universidade Federal de Mato Grosso do Sul

Algoritmos e Programação II, Análise de Sistemas, 2010

Conteúdo da aula

Introdução

- lista linear especial
- às vezes precisamos de manter um ponteiro para uma célula e também para a célula anterior para realizar algumas operações sobre a lista
- podemos precisar percorrer a lista linear nos dois sentidos
- para solucionar esse problema, adicionamos um novo campo ponteiro nas células da lista, que aponta para a célula anterior da lista

Introdução

- lista linear especial
- às vezes precisamos de manter um ponteiro para uma célula e também para a célula anterior para realizar algumas operações sobre a lista
- podemos precisar percorrer a lista linear nos dois sentidos
- para solucionar esse problema, adicionamos um novo campo ponteiro nas células da lista, que aponta para a célula anterior da lista

Introdução

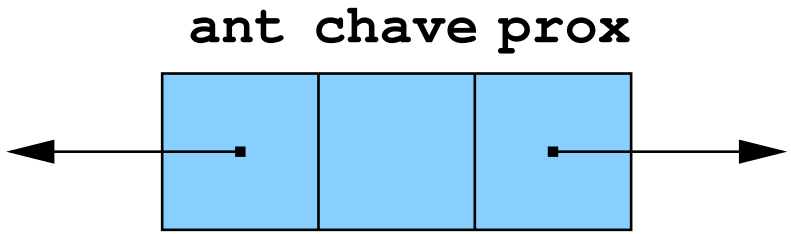
- lista linear especial
- às vezes precisamos de manter um ponteiro para uma célula e também para a célula anterior para realizar algumas operações sobre a lista
- podemos precisar percorrer a lista linear nos dois sentidos
- para solucionar esse problema, adicionamos um novo campo ponteiro nas células da lista, que aponta para a célula anterior da lista

Introdução

- lista linear especial
- às vezes precisamos de manter um ponteiro para uma célula e também para a célula anterior para realizar algumas operações sobre a lista
- podemos precisar percorrer a lista linear nos dois sentidos
- para solucionar esse problema, adicionamos um novo campo ponteiro nas células da lista, que aponta para a célula anterior da lista

Definição

- células de uma lista linear duplamente encadeada são ligadas por ponteiros que indicam a posição da célula anterior e da próxima célula da lista



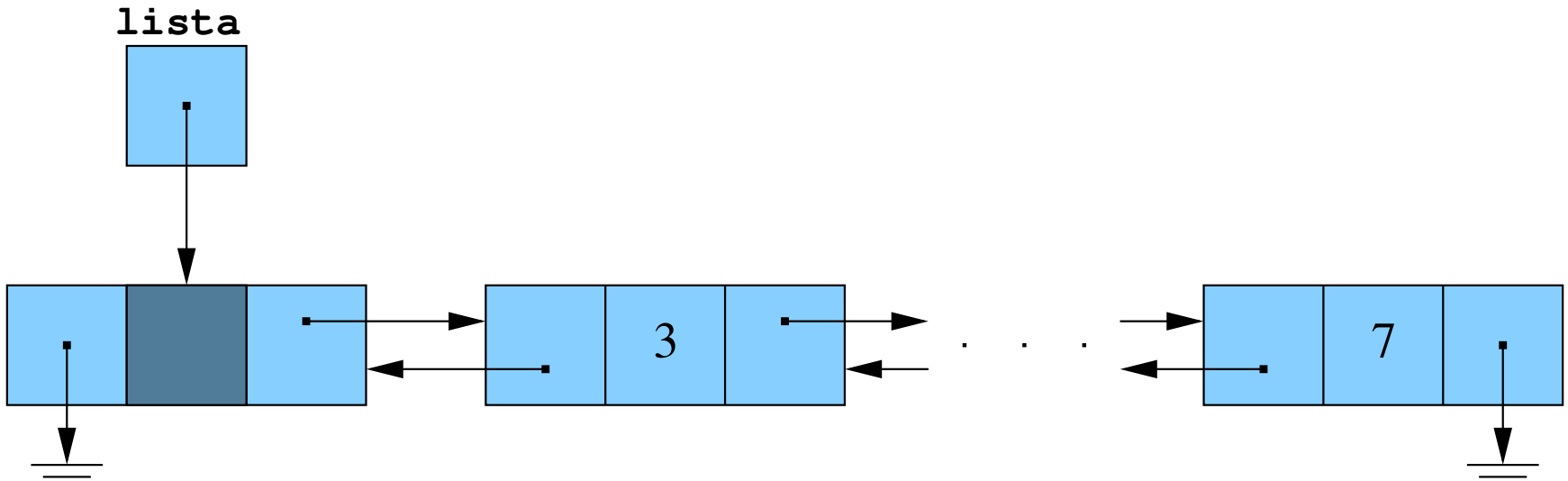
Definição

- uma célula é definida como

```
typedef struct cel {
    int chave;
    struct cel *ant;
    struct cel *prox;
} celula;
```

Definição

- um exemplo de uma lista linear duplamente encadeada



Definição

- declaração e inicialização de uma lista linear duplamente encadeada com cabeça

```
celula *lista;
lista = (celula *) malloc(sizeof (celula));
lista->ant = NULL;
lista->prox = NULL;
```

- declaração e inicialização de uma lista linear duplamente encadeada sem cabeça

```
celula *lista;
lista = NULL;
```

Definição

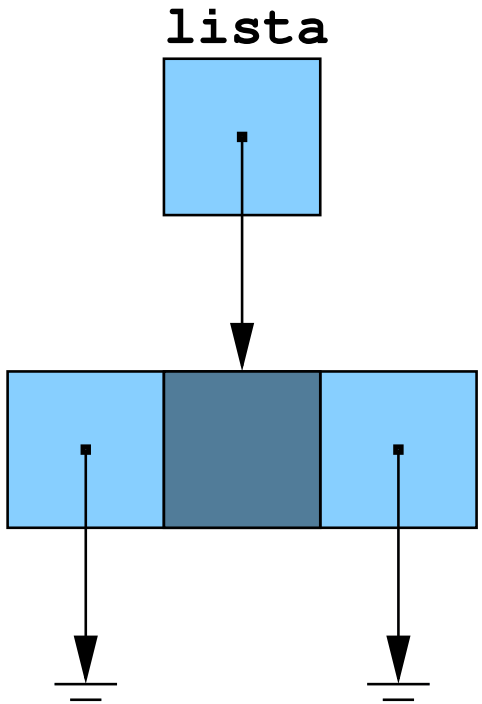
- declaração e inicialização de uma lista linear duplamente encadeada com cabeça

```
celula *lista;
lista = (celula *) malloc(sizeof (celula));
lista->ant = NULL;
lista->prox = NULL;
```

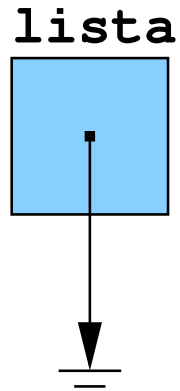
- declaração e inicialização de uma lista linear duplamente encadeada sem cabeça

```
celula *lista;
lista = NULL;
```

Definição



(a)



(b)

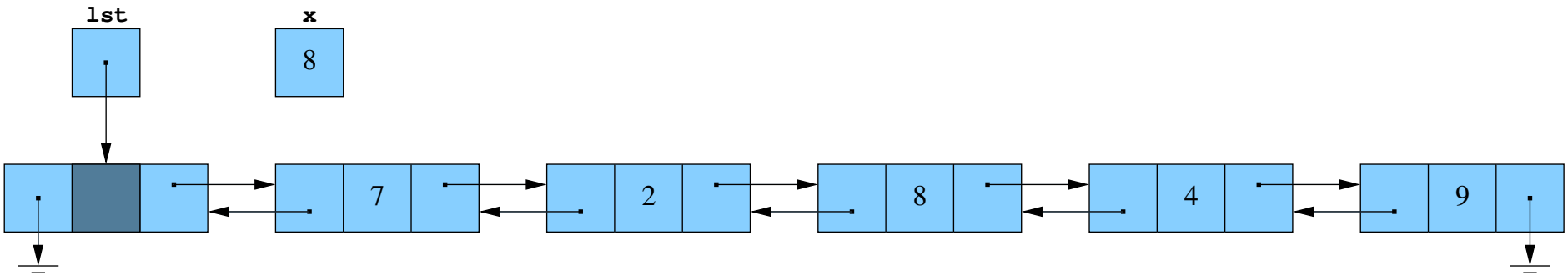
Busca

```
celula *busca_dup_C(int x, celula *lst)
{
   celula *p;

   p = lst->prox;
   while (p != NULL && p->chave != x)
      p = p->prox;

   return p;
}
```

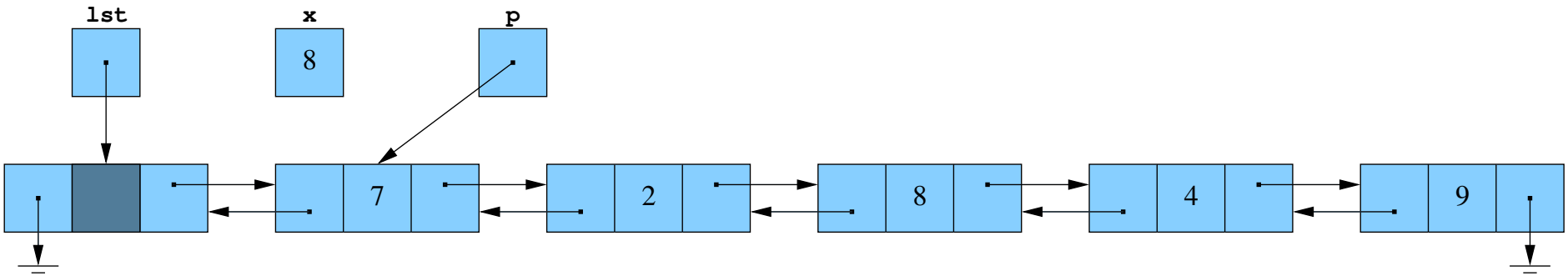
Busca seguida de remoção

```
void busca_remove_dup_C(int x, celula *lst)
{
   celula *p;

   p = lst->prox;
   while (p != NULL && p->chave != x)
      p = p->prox;
   if (p != NULL) {
      p->ant->prox = p->prox;
      if (p->prox != NULL)
         p->prox->ant = p->ant;
      free(p);
   }
}
```

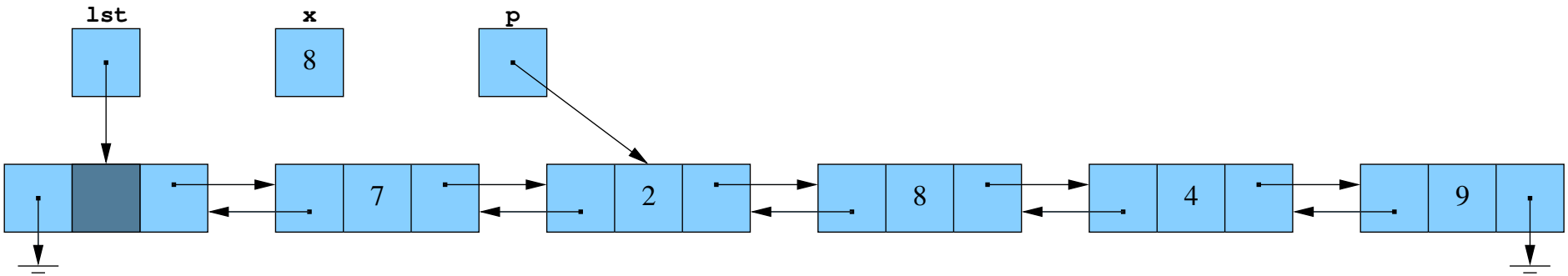
Busca seguida de remoção



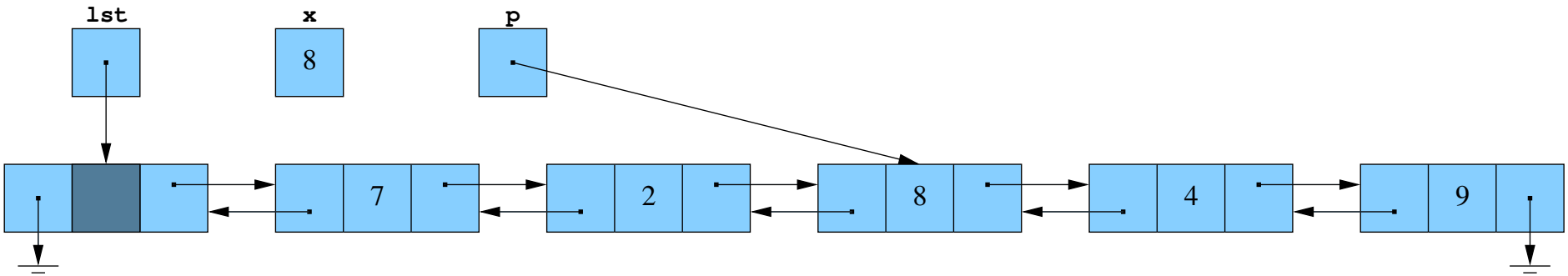
Busca seguida de remoção



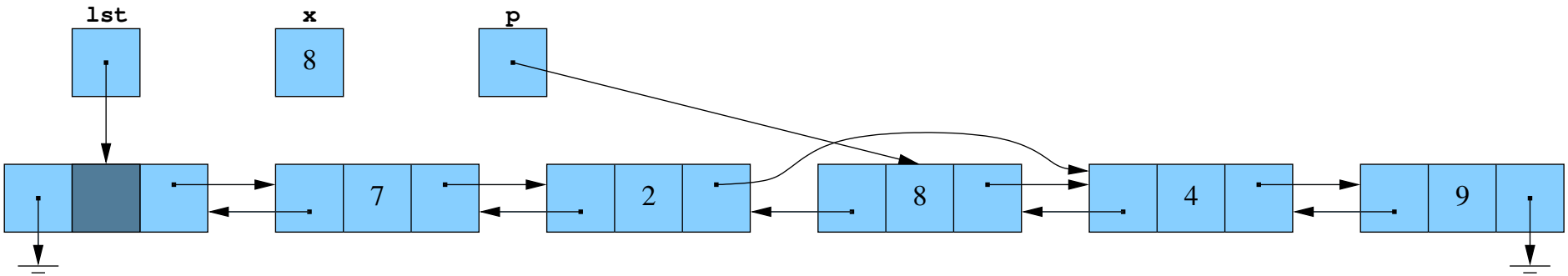
Busca seguida de remoção



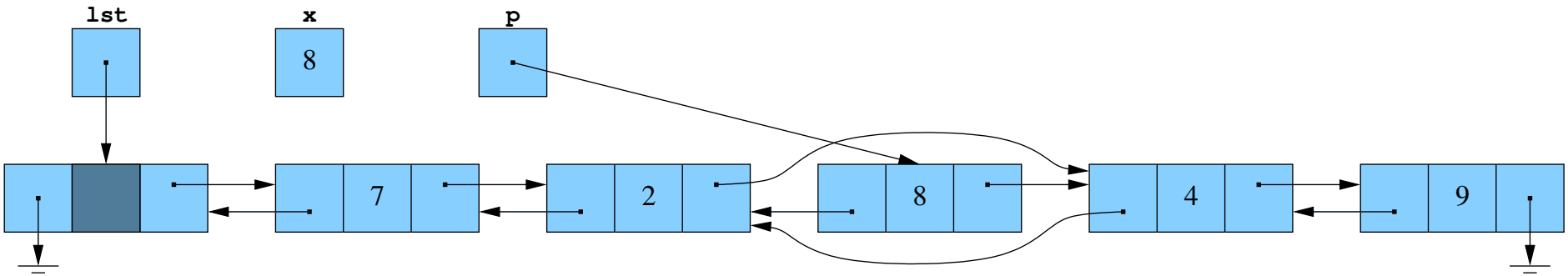
Busca seguida de remoção



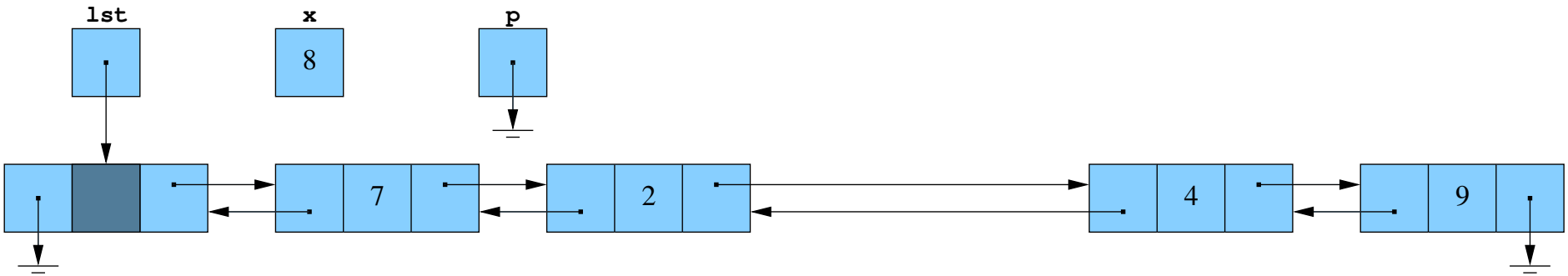
Busca seguida de remoção



Busca seguida de remoção



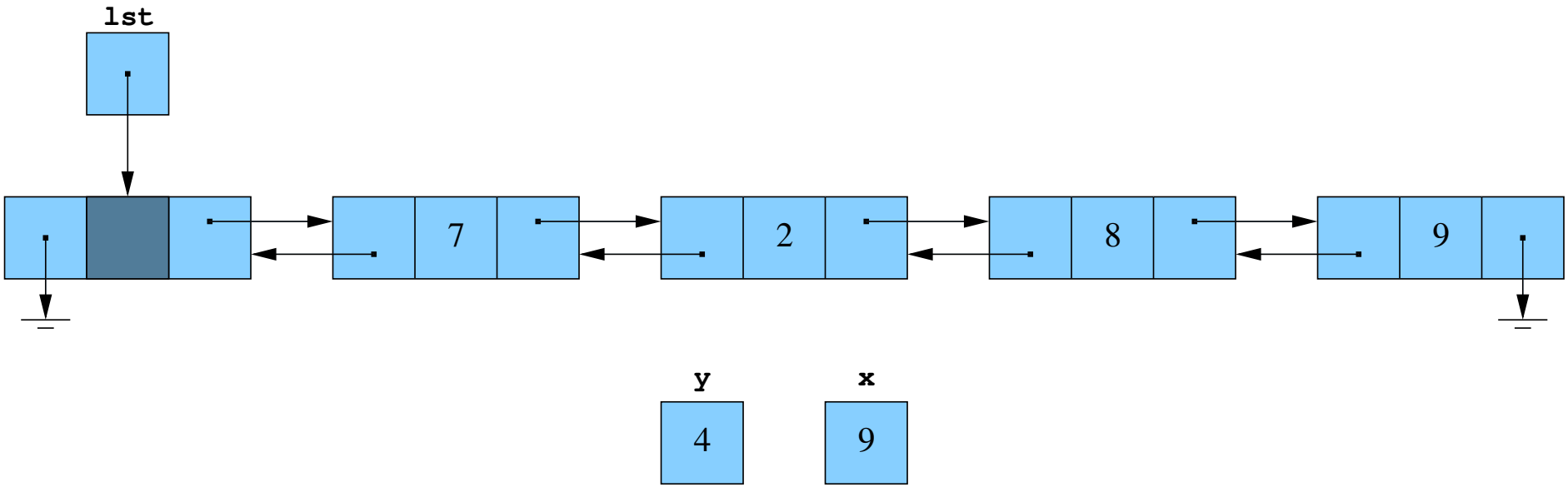
Busca seguida de remoção



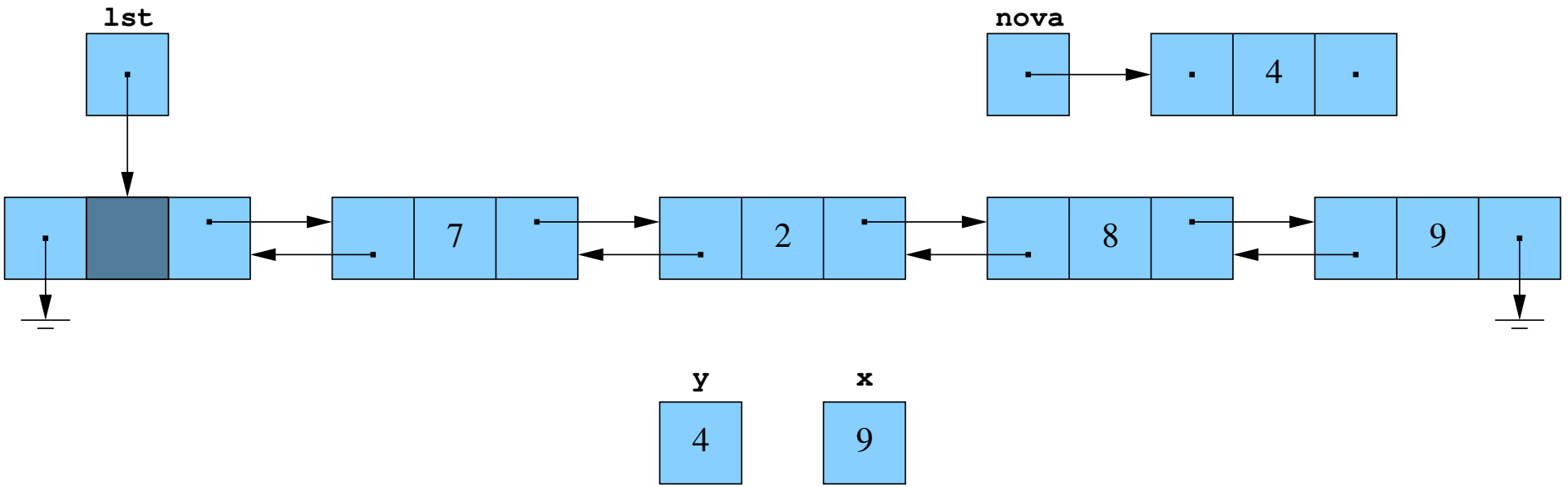
Busca seguida de inserção

```
void busca_insere_dup_C(int y, int x, celula *lst)
{
   celula *p, *q, *nova;

   nova = (celula *) malloc(sizeof (celula));
   nova->chave = y;
   p = lst;
   q = lst->prox;
   while (q != NULL && q->chave != x) {
      p = q;
      q = q->prox;
   }
   nova->ant = p;
   nova->prox = q;
   if (p != NULL)
      p->prox = nova;
   if (q != NULL)
      q->ant = nova;
}
```

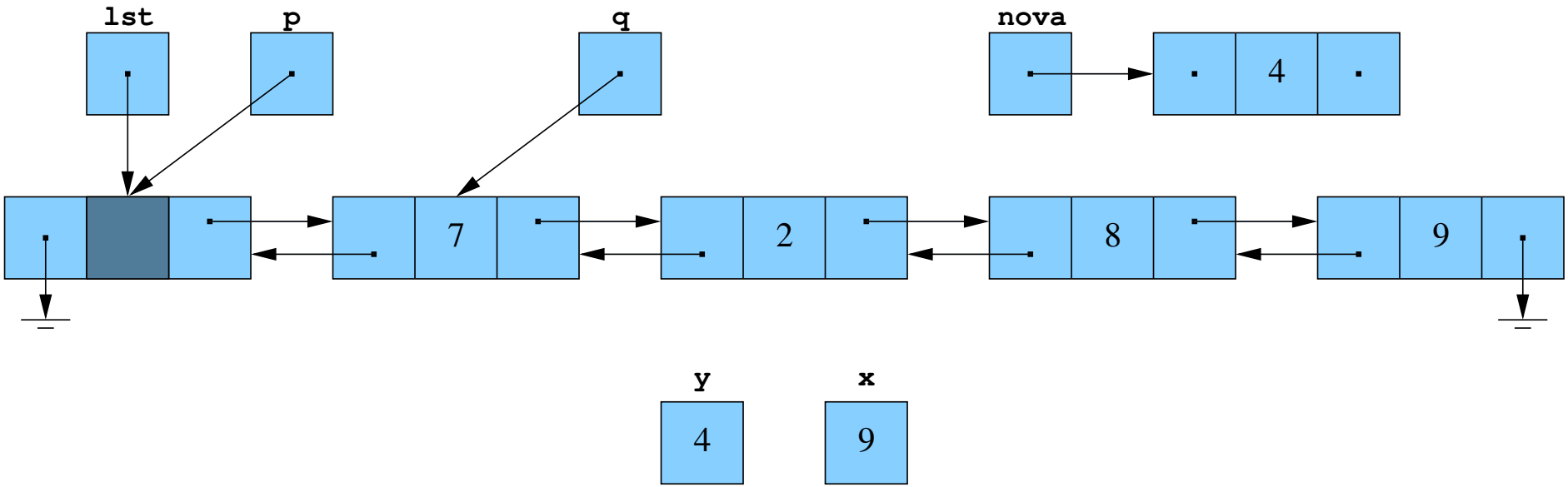
Busca seguida de inserção



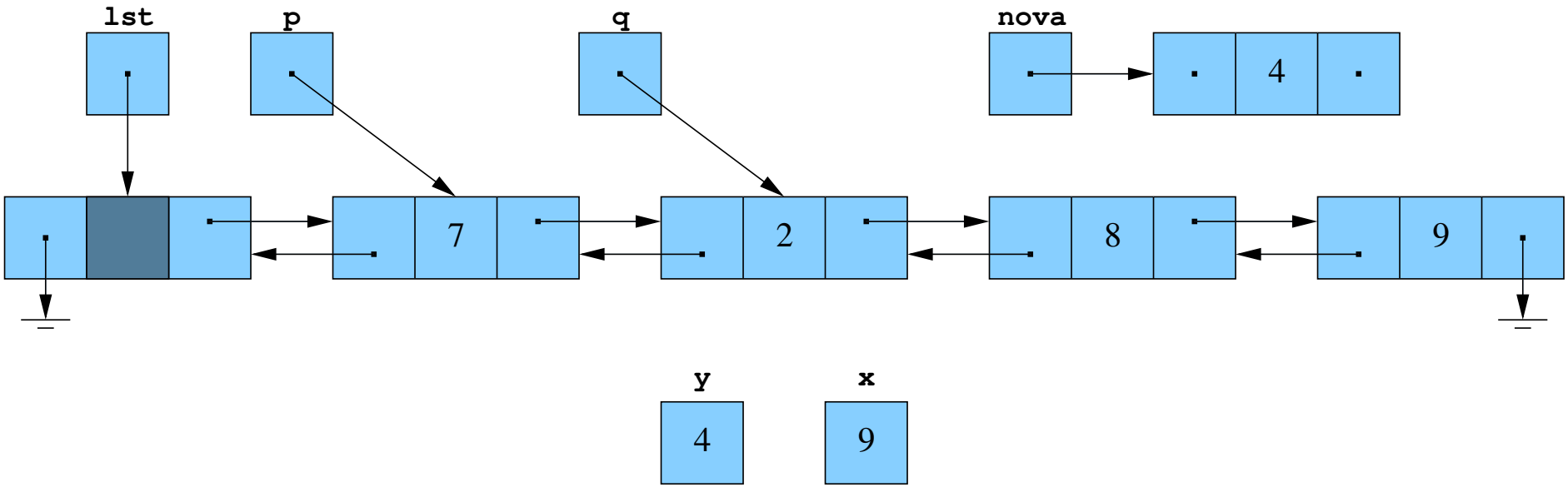
Busca seguida de inserção



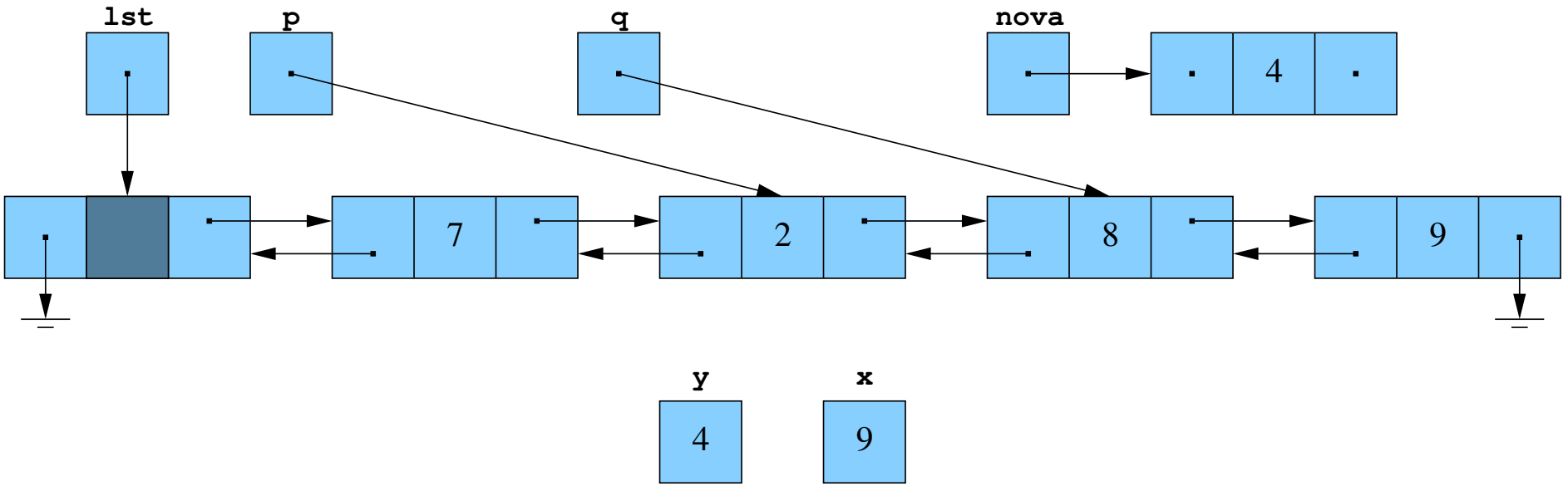
Busca seguida de inserção



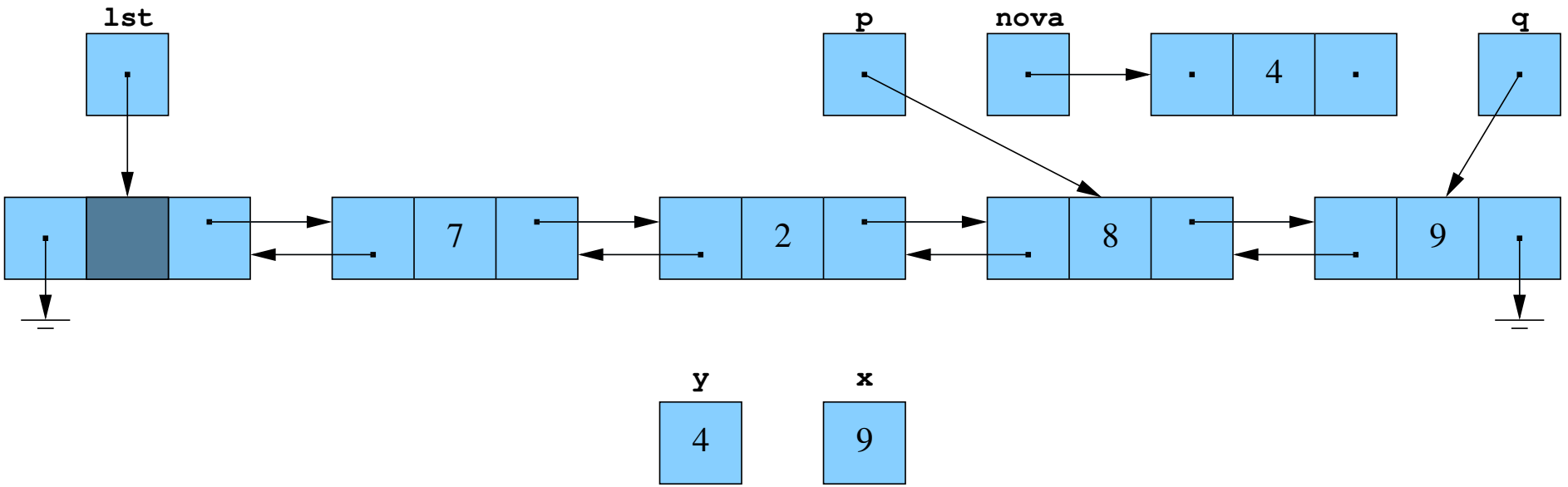
Busca seguida de inserção



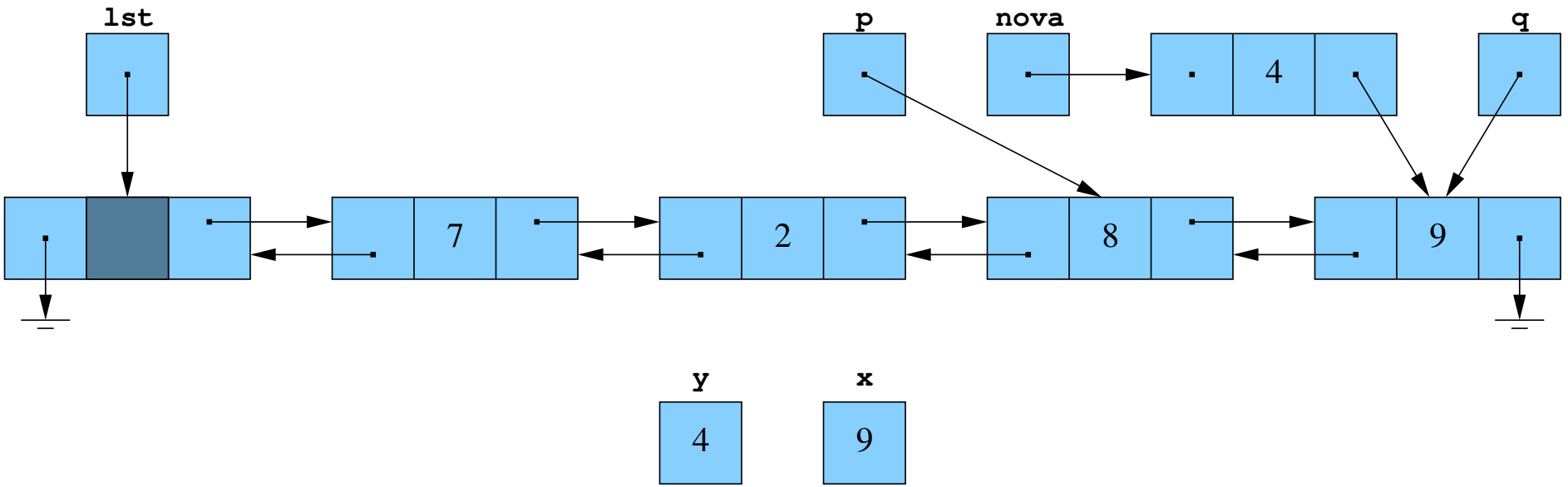
Busca seguida de inserção



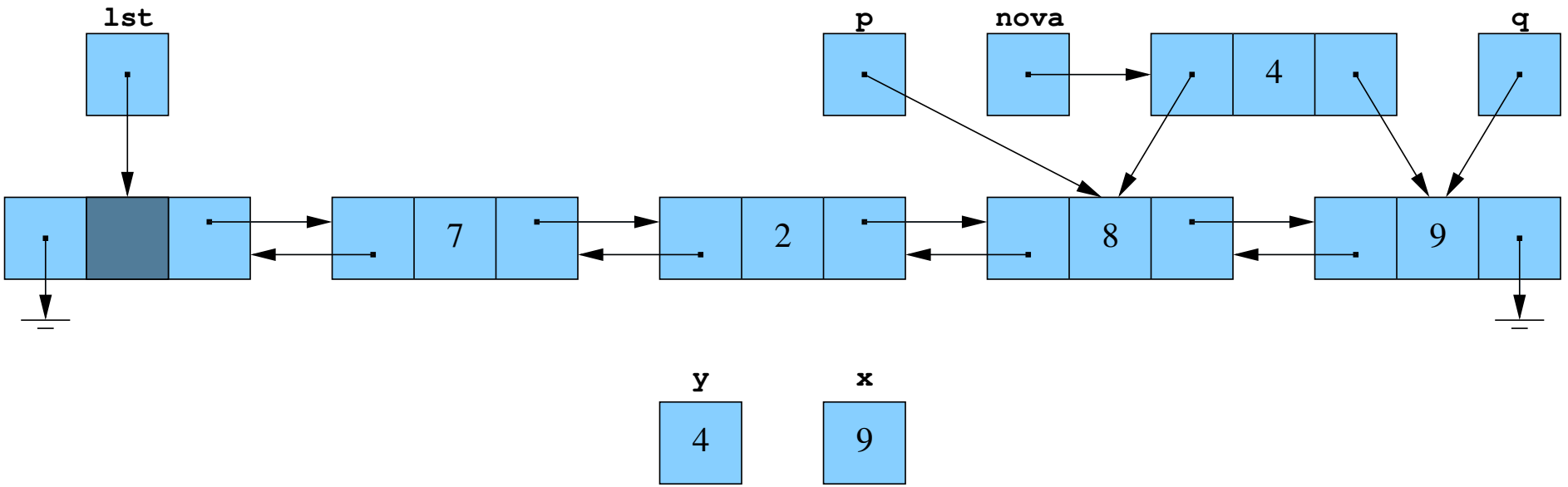
Busca seguida de inserção



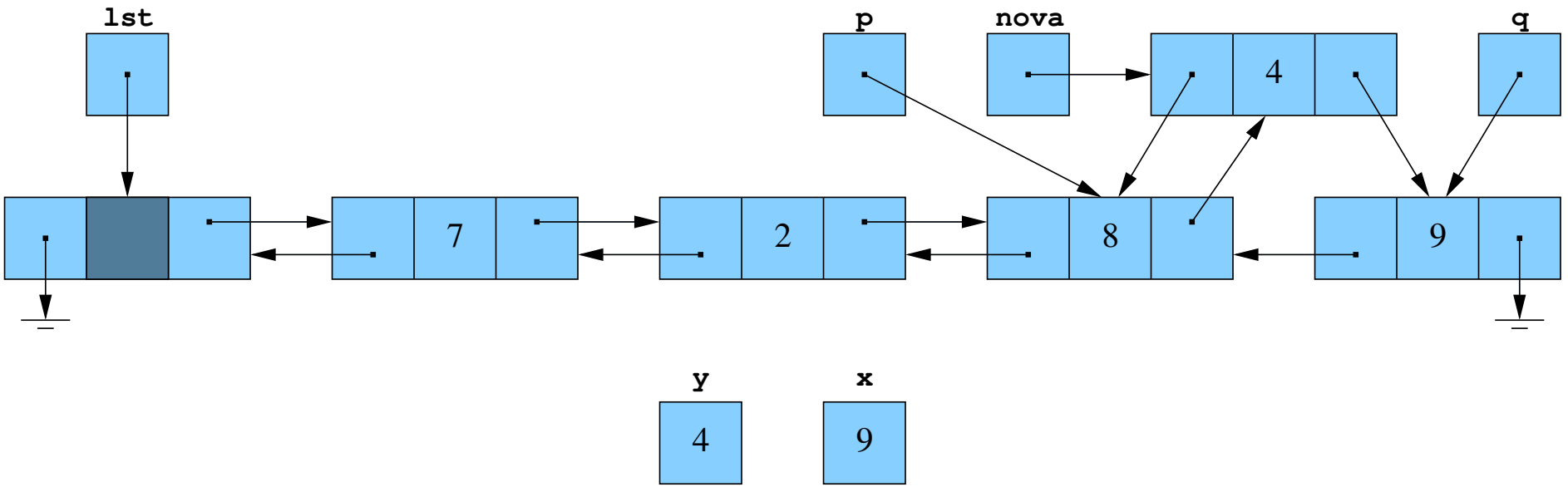
Busca seguida de inserção



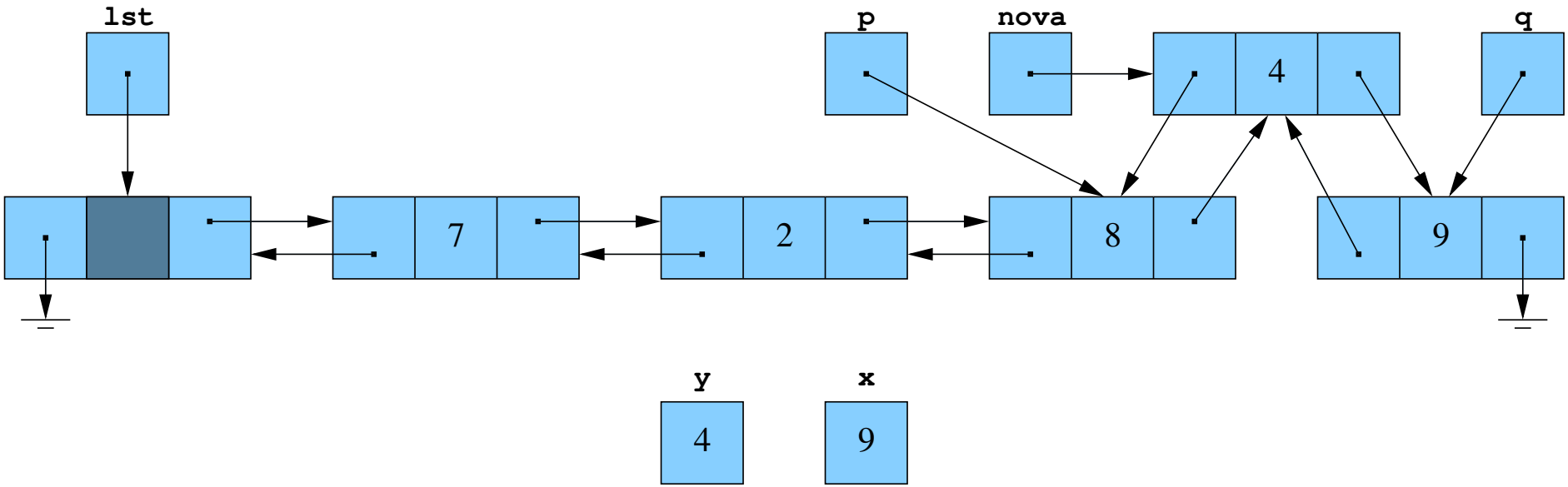
Busca seguida de inserção



Busca seguida de inserção



Busca seguida de inserção



Exercícios

22.1 Escreva funções para implementar as operações básicas de busca, inserção e remoção em uma lista linear duplamente encadeada sem cabeça

22.2 Escreva uma função que receba um ponteiro para o início de uma lista linear em alocação duplamente encadeada, um ponteiro para o fim dessa lista e uma chave, e realize a inserção dessa chave no final da lista

Exercícios

22.1 Escreva funções para implementar as operações básicas de busca, inserção e remoção em uma lista linear duplamente encadeada sem cabeça

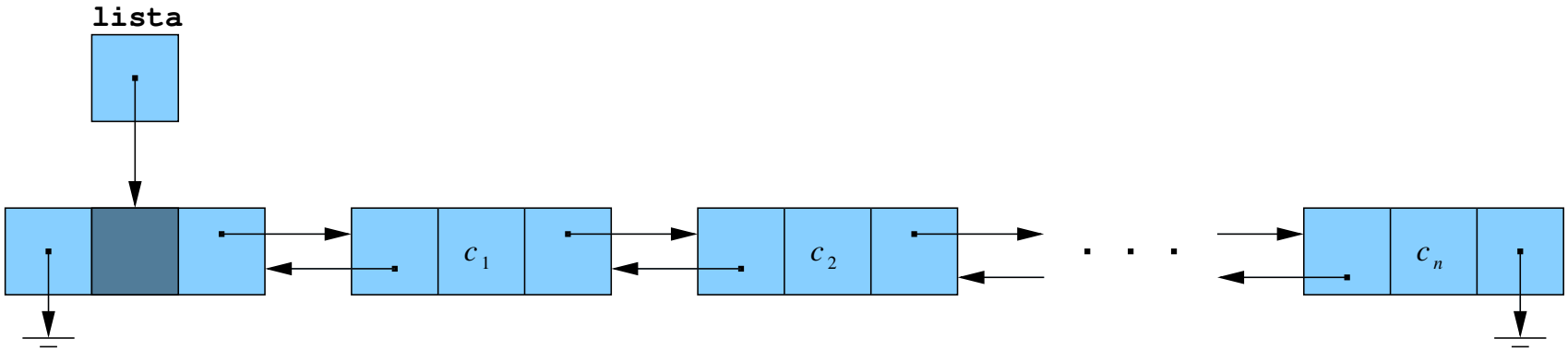
22.2 Escreva uma função que receba um ponteiro para o início de uma lista linear em alocação duplamente encadeada, um ponteiro para o fim dessa lista e uma chave, e realize a inserção dessa chave no final da lista

Exercícios

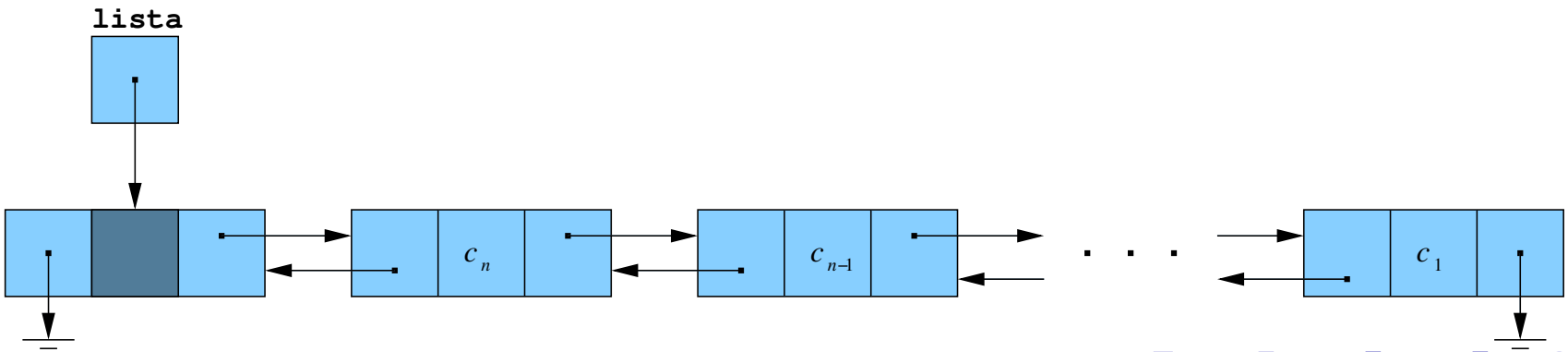
22.3 Listas lineares duplamente encadeadas também podem ser listas circulares. Basta olhar para uma lista linear duplamente encadeada e fazer o ponteiro para a céula anterior da célula cabeça apontar para a última célula e a ponteiro para a próxima célula da última célula apontar para a célula cabeça. Escreva funções para implementar as operações básicas de busca, inserção e remoção em uma lista linear duplamente encadeada circular com cabeça

Exercícios

22.4 Considere uma lista linear duplamente encadeada contendo as chaves $c_1, c_2, \ldots, c_n$, como na figura abaixo.



Escreva uma função que altere os ponteiros `ant` e `prox` da lista, *sem* mover suas informações, tal que a lista fique invertida como na figura a seguir.



Exercícios

22.5 Suponha que você queira manter uma lista linear duplamente encadeada com cabeça com as chaves em ordem crescente. Escreva as funções de busca, inserção e remoção para essa lista