Listas encadeadas

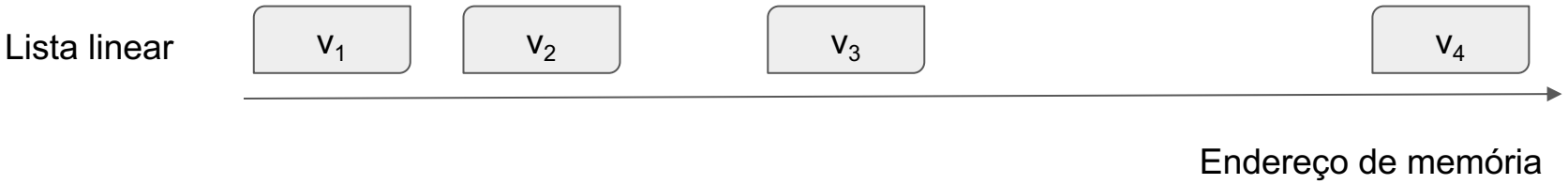
Prof. Martín Vigil

Definição de Lista

- É uma ***sequência*** de dados
- Exemplos:
  - <Pedro, João, Maria, Ivan>
  - <5, 9, 70, 70, 3>
  - <😂, 🙃, 🧐, 👻>

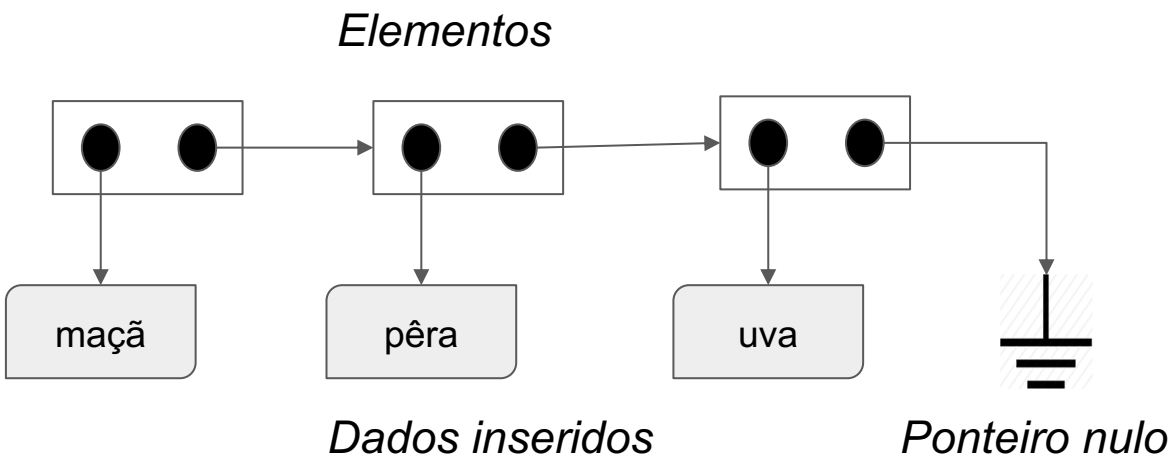
Listas lineares

- Como vetores, são sequências de dados <d_1, d_2,…, d_n>
- Lineares pois cada dado tem no máximo um sucessor
- Porém,
  - Os dados podem ocupar regiões não contíguas da memória
  - A lista tem capacidade (teoricamente) ilimitada



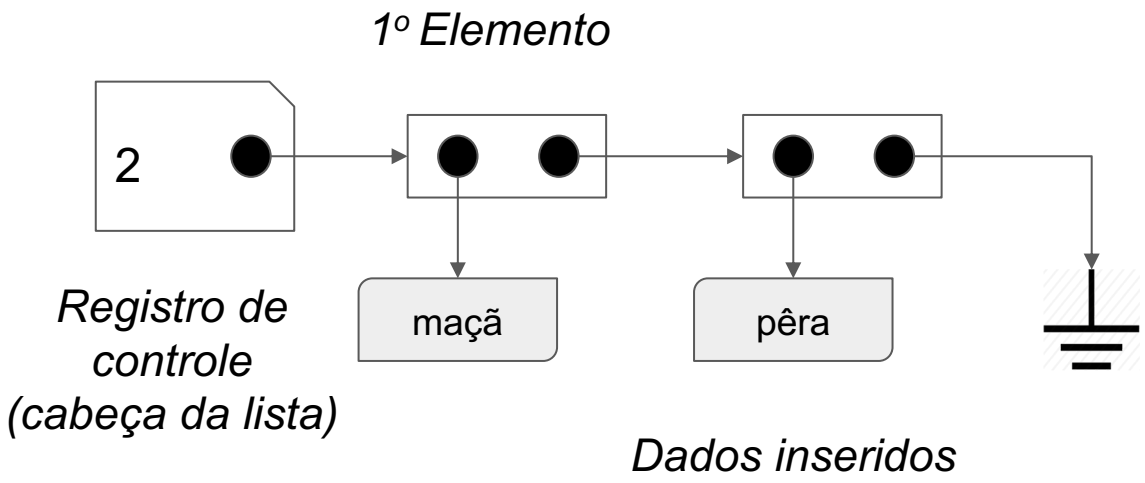
Listas encadeadas simples

- São um caso de lista linear
- Consistem de objetos (instâncias) do tipo ***Elemento*** e ponteiros
- Cada elemento
  - é alocado ou desalocado dinamicamente quando necessário
  - aponta para um dado inserido na lista
  - aponta para o próximo elemento na lista (possivelmente nulo)
- São percorridas somente no sentido dos ponteiros



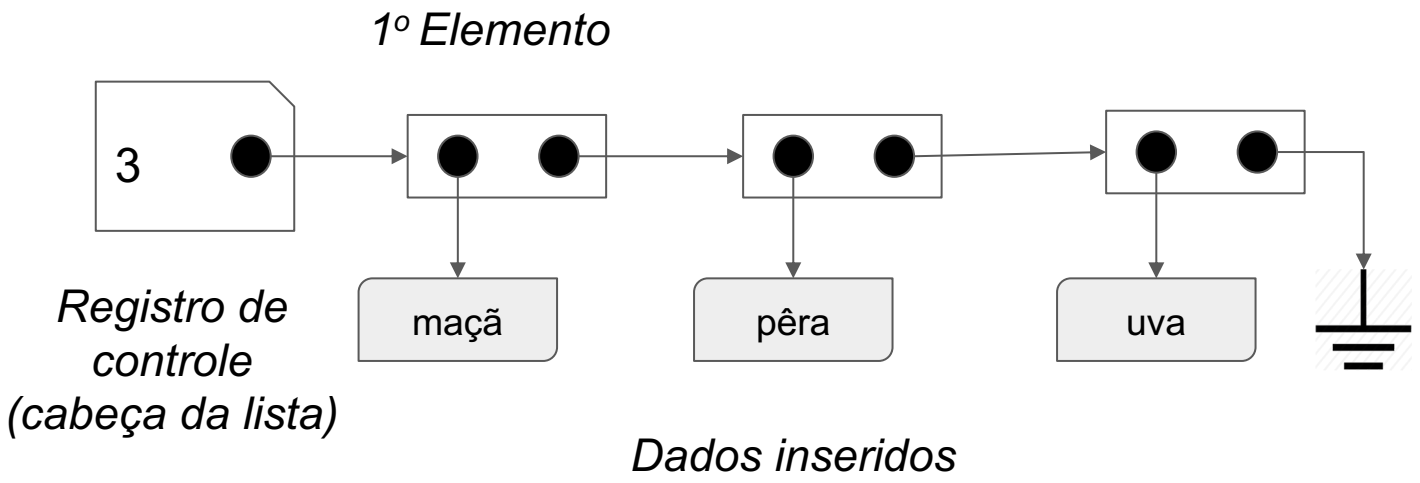
Detalhando a lista encadeada simples

- Elementos são registros
- Um registro de controle é necessário para identificar
  - a quantidade de elementos que a lista encadeada simples contém
  - o primeiro elemento da lista



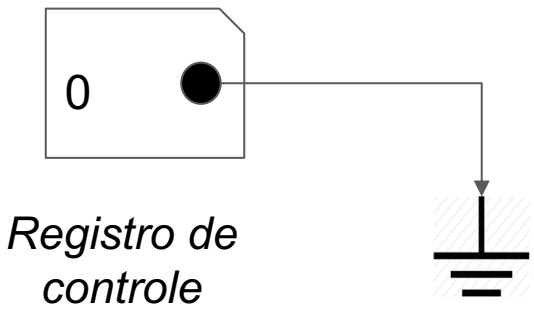
Detalhando a lista encadeada simples

- Elementos são registros
- Um registro de controle é necessário para identificar
  - a quantidade de elementos que a lista encadeada simples contém
  - o primeiro elemento da lista



Detalhando a lista encadeada simples

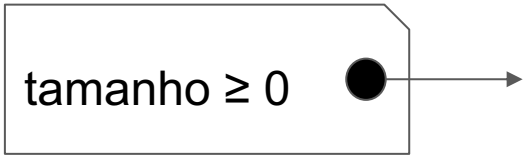
- Elementos são registros
- Um registro de controle é necessário para identificar
  - a quantidade de elementos que a lista encadeada simples contém
  - o primeiro elemento da lista



Modelagem do registro de controle

- Aspecto estrutural:
  - Precisamos de um ponteiro para o primeiro elemento
  - Precisamos de um número (natural) para indicar a quantidade de elementos na lista (tamanho da lista)

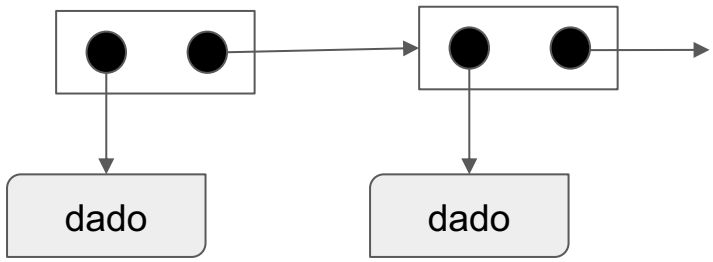
```
registro ListaEncSimples {
    Elemento* primeiro
    Número tamanho
}
```



Modelagem do registro de elemento

- Aspecto estrutural
  - Precisamos de um ponteiro para o próximo elemento (possivelmente nulo)
  - Precisamos de um ponteiro para o dado T *genérico* inserido na lista

```
registro Elemento {
    T*  dado
    Elemento* próximo
}
```

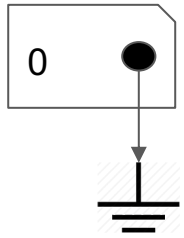


Operações comuns sobre listas encadeadas simples

1. Criar vazia (inicialmente vazia)
2. Inserir ou remover dados de uma posição da lista
3. Obter o dado em uma posição da lista
4. Obter a posição de um dado na lista
5. Destruir lista

Criar lista vazia: passos

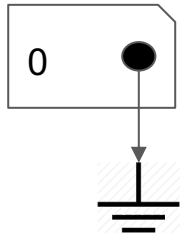
1. Aloca-se espaço em memória para o registro ListaEncSimples
2. Iniciam-se as variáveis internas
   a) O tamanho deve ser zero
   b) O ponteiro para o primeiro elemento deve apontar para NULO



Criar lista vazia: passos

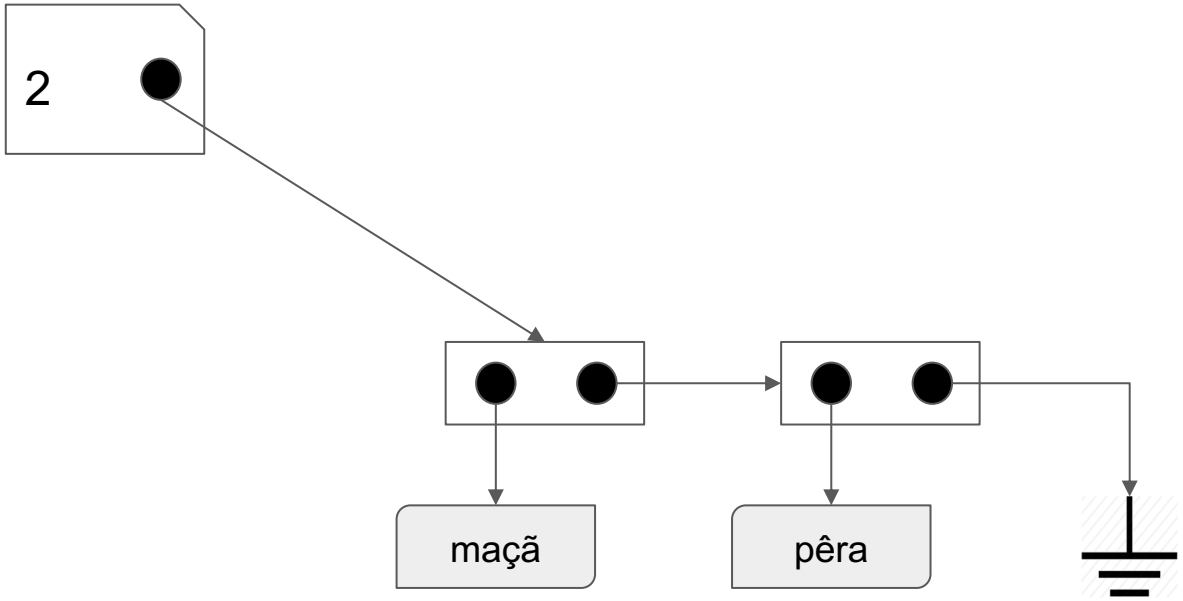
1. Aloca-se espaço em memória para o registro ListaEncSimples
2. Iniciam-se as variáveis internas
   a) O tamanho deve ser zero
   b) O ponteiro para o primeiro elemento deve apontar para NULO

Custo computacional: O(1)



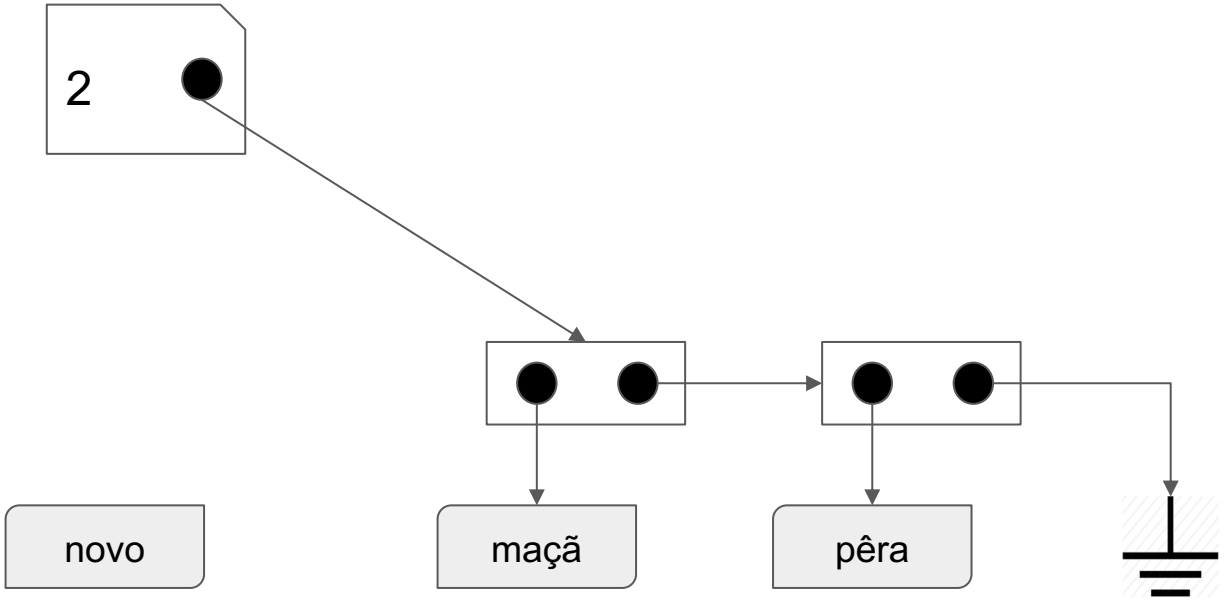
Inserir um dado no **início da lista**: passos

1. Deve-se alocar um novo elemento
2. Este elemento deve apontar para o dado e o primeiro elemento da lista
3. O registro de controle deve apontar para o novo elemento
4. O tamanho da lista deve ser incrementado



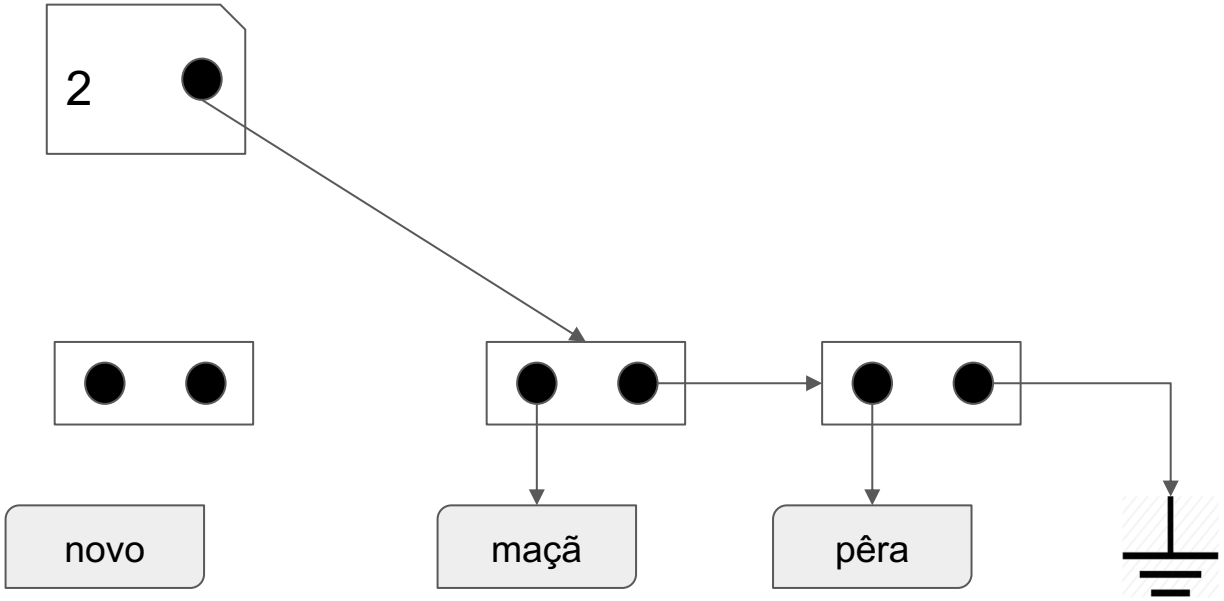
Inserir um dado no início da lista: passos

1. Deve-se alocar um novo elemento
2. Este elemento deve apontar para o dado e o primeiro elemento da lista
3. O registro de controle deve apontar para o novo elemento
4. O tamanho da lista deve ser incrementado



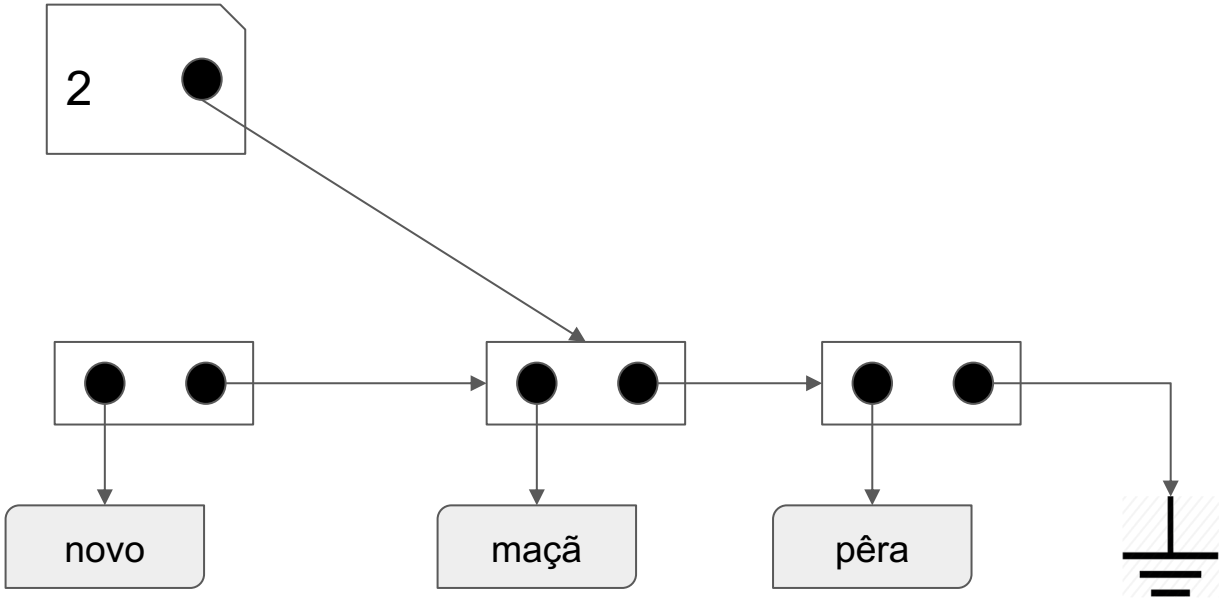
Inserir um dado no início da lista: passos

1. Deve-se alocar um novo elemento
2. Este elemento deve apontar para o dado e o primeiro elemento da lista
3. O registro de controle deve apontar para o novo elemento
4. O tamanho da lista deve ser incrementado



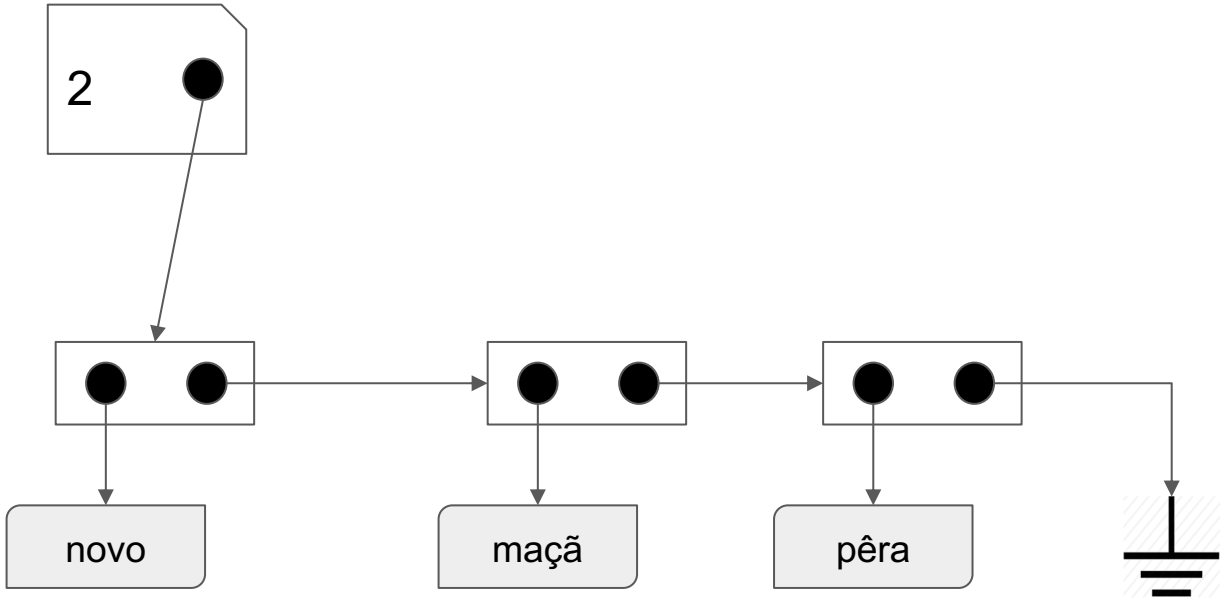
Inserir um dado no início da lista: passos

1. Deve-se alocar um novo elemento
2. Este elemento deve apontar para o dado e o primeiro elemento da lista
3. O registro de controle deve apontar para o novo elemento
4. O tamanho da lista deve ser incrementado



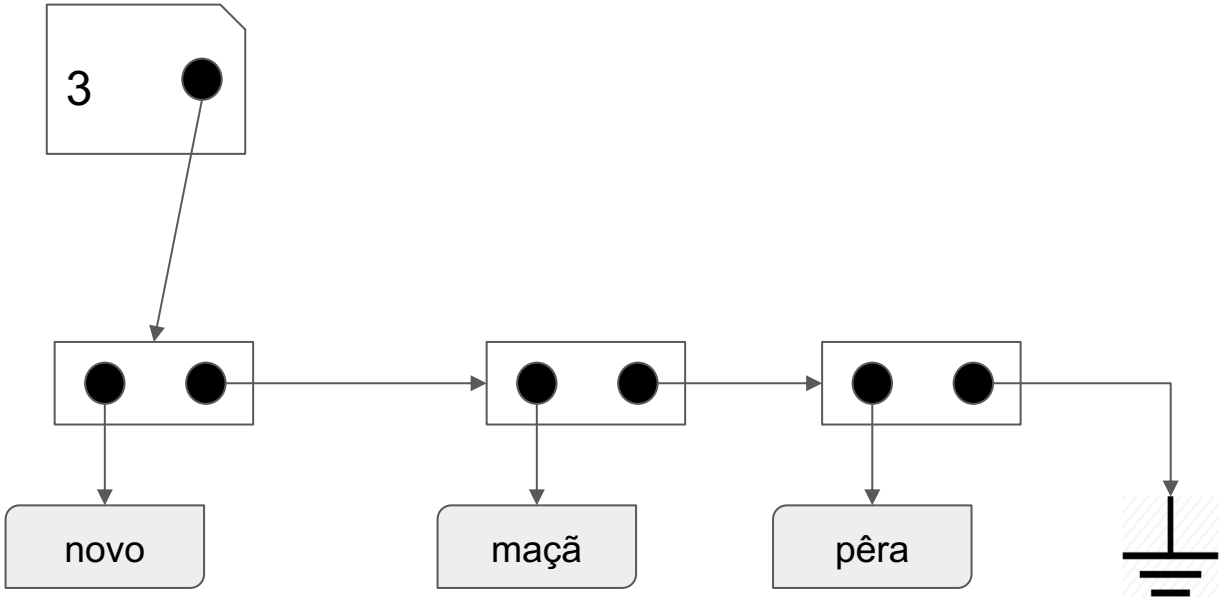
Inserir um dado no início da lista: passos

1. Deve-se alocar um novo elemento
2. Este elemento deve apontar para o dado e o primeiro elemento da lista
3. O registro de controle deve apontar para o novo elemento
4. O tamanho da lista deve ser incrementado



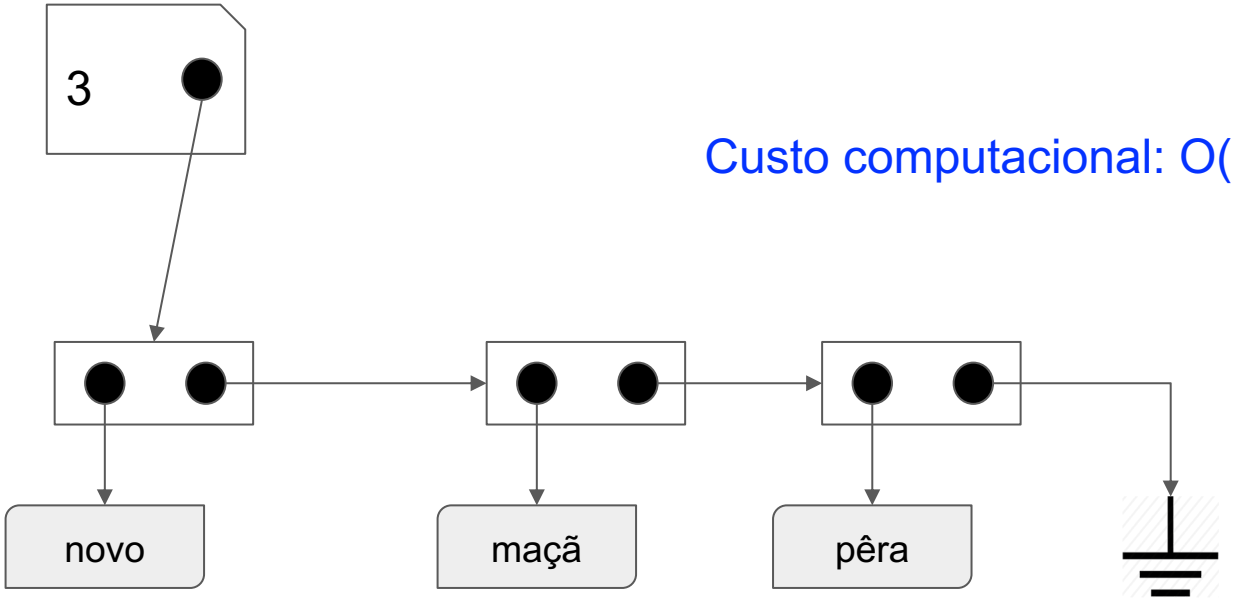
Inserir um dado no início da lista: passos

1. Deve-se alocar um novo elemento
2. Este elemento deve apontar para o dado e o primeiro elemento da lista
3. O registro de controle deve apontar para o novo elemento
4. O tamanho da lista deve ser incrementado



Inserir um dado no início da lista: passos

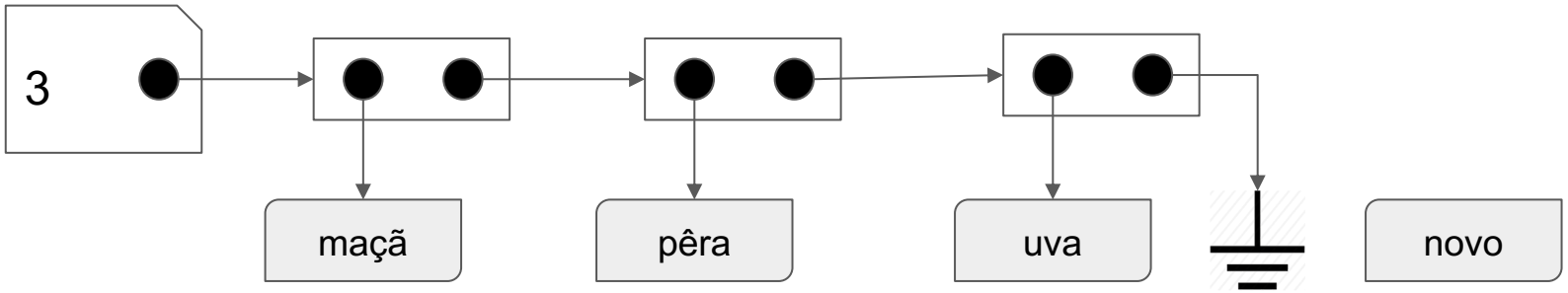
1. Deve-se alocar um novo elemento
2. Este elemento deve apontar para o dado e o primeiro elemento da lista
3. O registro de controle deve apontar para o novo element
4. O tamanho da lista deve ser incrementado



Custo computacional: O(1)

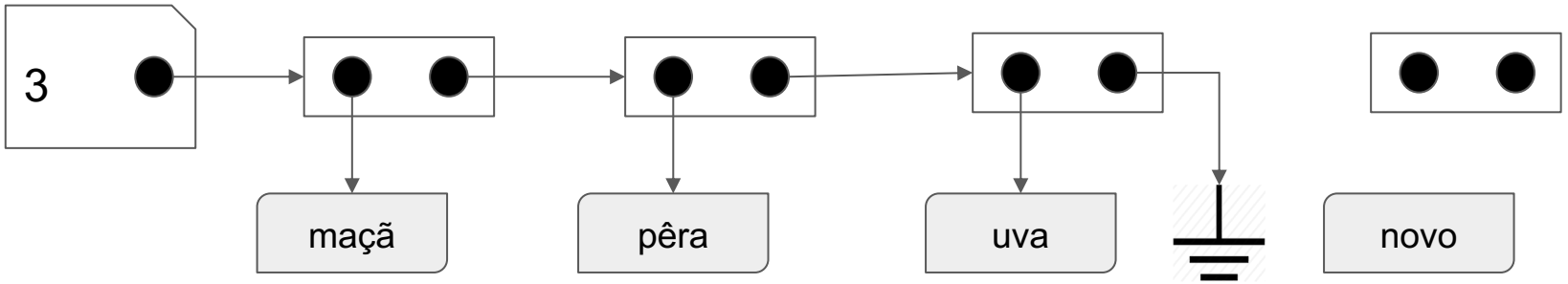
Inserir um dado no fim da lista: passos

1. Deve-se alocar um novo elemento
2. Este elemento deve apontar para o dado e NULO
3. O último elemento da lista deve apontar para o novo elemento
4. O tamanho da lista deve ser incrementado



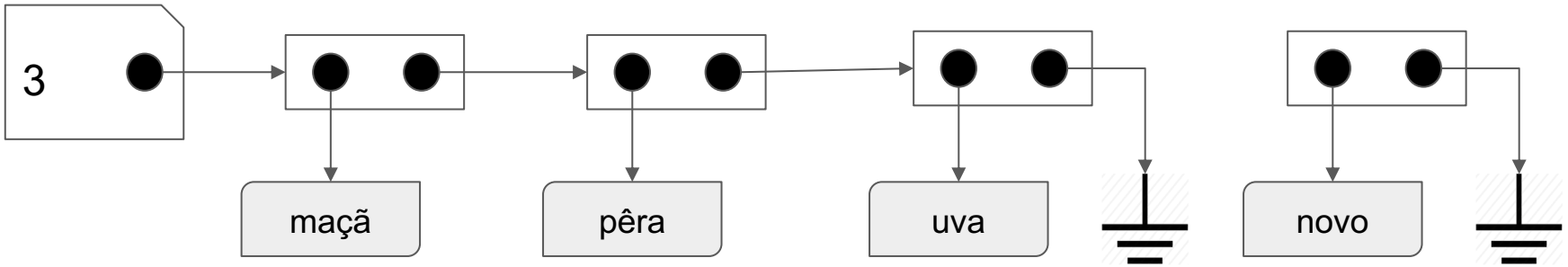
Inserir um dado no fim da lista: passos

1. Deve-se alocar um novo elemento
2. Este elemento deve apontar para o dado e NULO
3. O último elemento da lista deve apontar para o novo elemento
4. O tamanho da lista deve ser incrementado



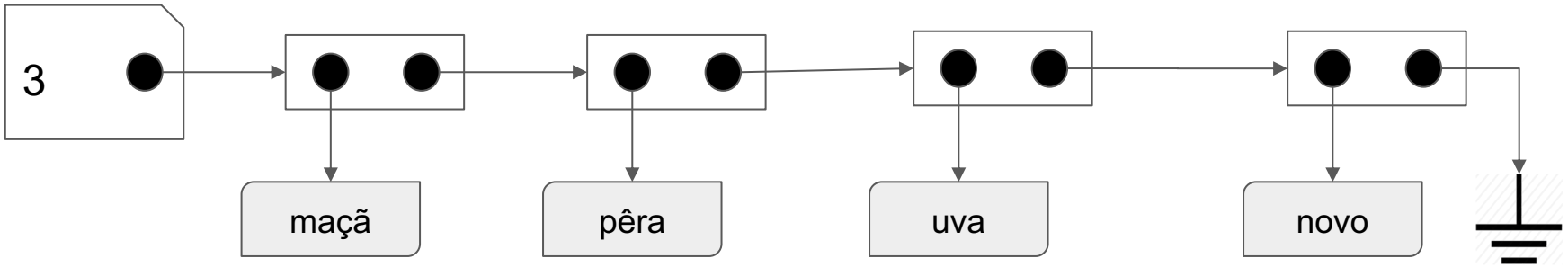
Inserir um dado no fim da lista: passos

1. Deve-se alocar um novo elemento
2. Este elemento deve apontar para o dado e NULO
3. O último elemento da lista deve apontar para o novo elemento
4. O tamanho da lista deve ser incrementado



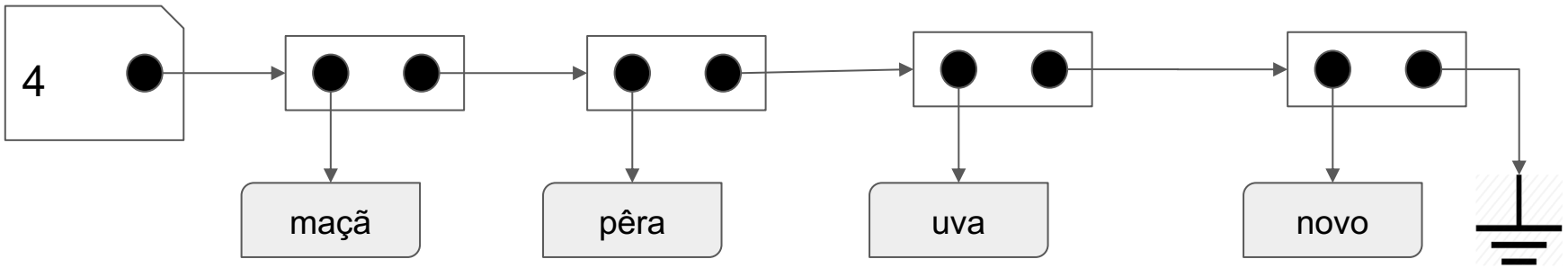
Inserir um dado no fim da lista: passos

1. Deve-se alocar um novo elemento
2. Este elemento deve apontar para o dado e NULO
3. O último elemento da lista deve apontar para o novo elemento
4. O tamanho da lista deve ser incrementado



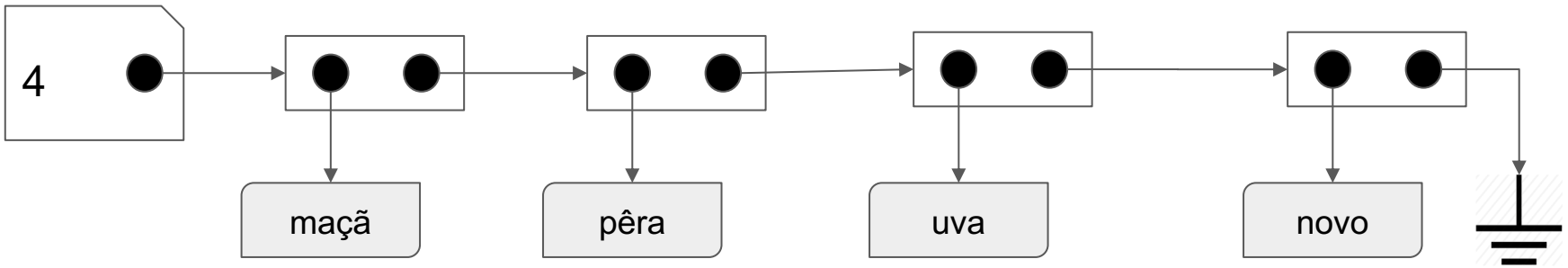
Inserir um dado no fim da lista: passos

1. Deve-se alocar um novo elemento
2. Este elemento deve apontar para o dado e NULO
3. O último elemento da lista deve apontar para o novo elemento
4. O tamanho da lista deve ser incrementado



Inserir um dado no fim da lista: passos

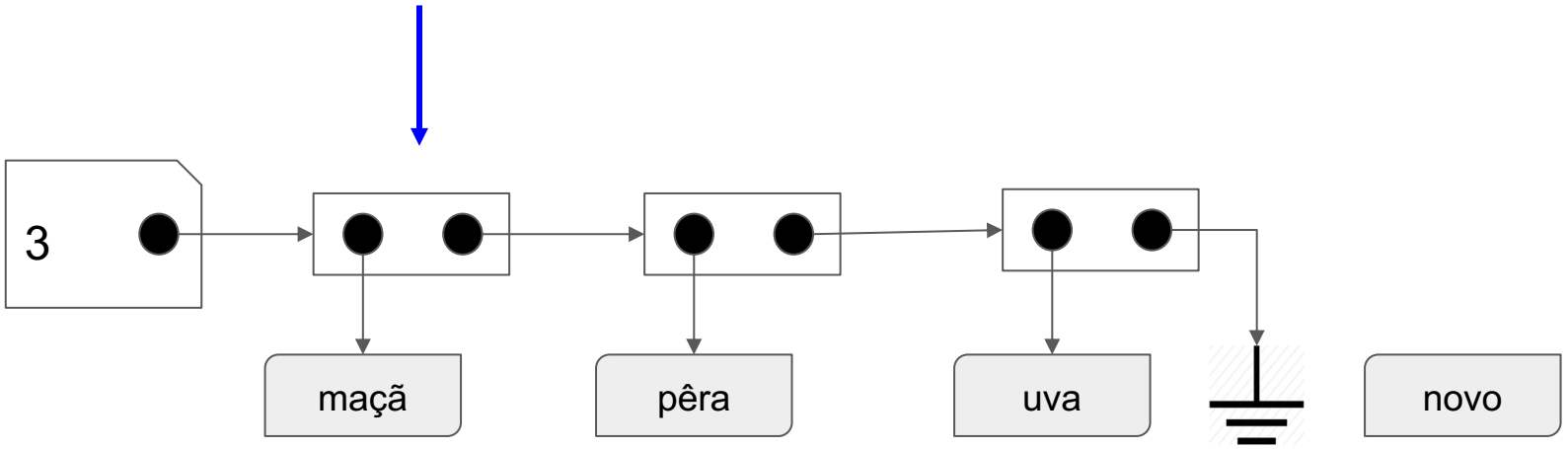
1. Deve-se alocar um novo elemento
2. Este elemento deve apontar para o dado e NULO
3. O último elemento da lista deve apontar para o novo element
4. O tamanho da lista deve ser incrementado



Custo computacional: O(1)?

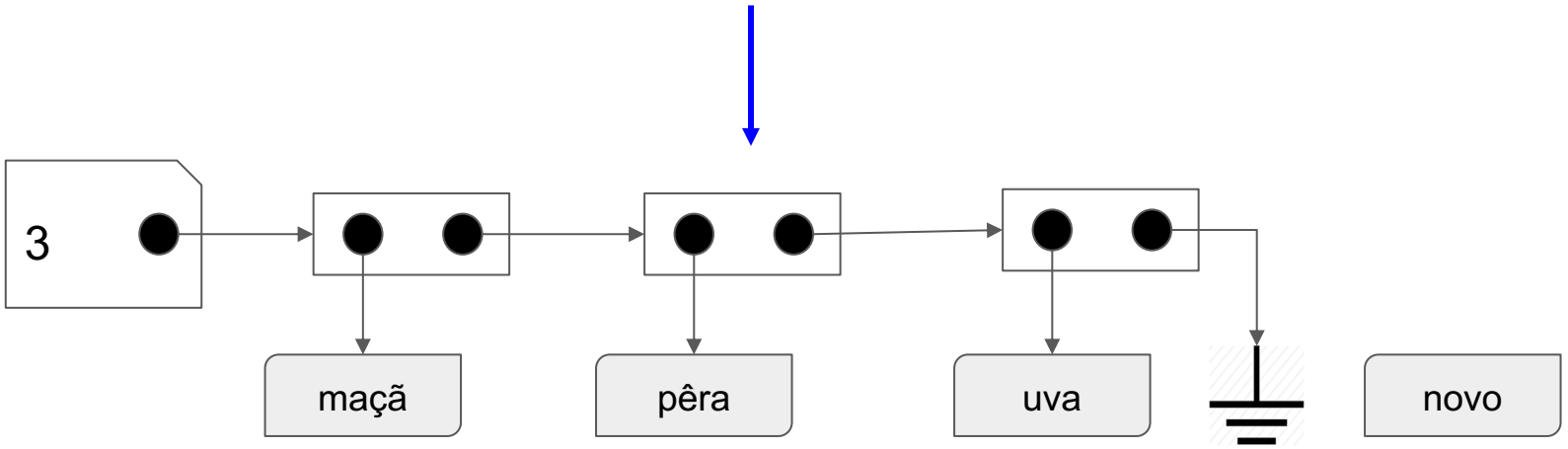
Inserir um dado no fim da lista: detalhes

- É preciso modificar o último elemento da lista para fazer a inserção
- Para isso, é preciso primeiro percorrer a lista até esse elemento



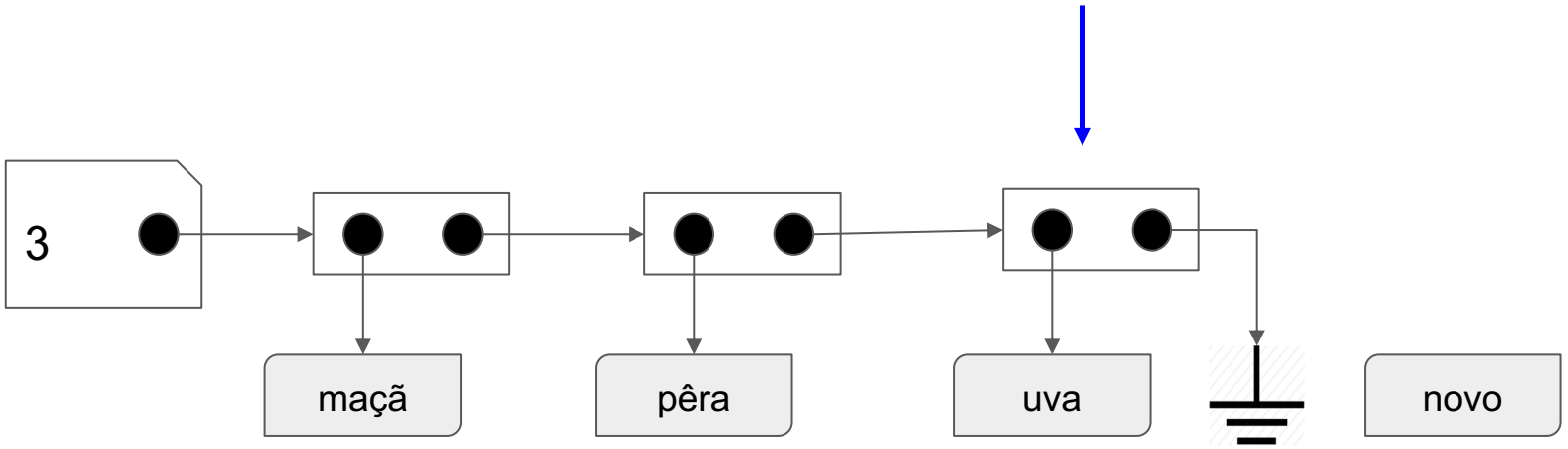
Inserir um dado no fim da lista: detalhes

- É preciso modificar o último elemento da lista para fazer a inserção
- Para isso, é preciso primeiro percorrer a lista até esse elemento



Inserir um dado no fim da lista: detalhes

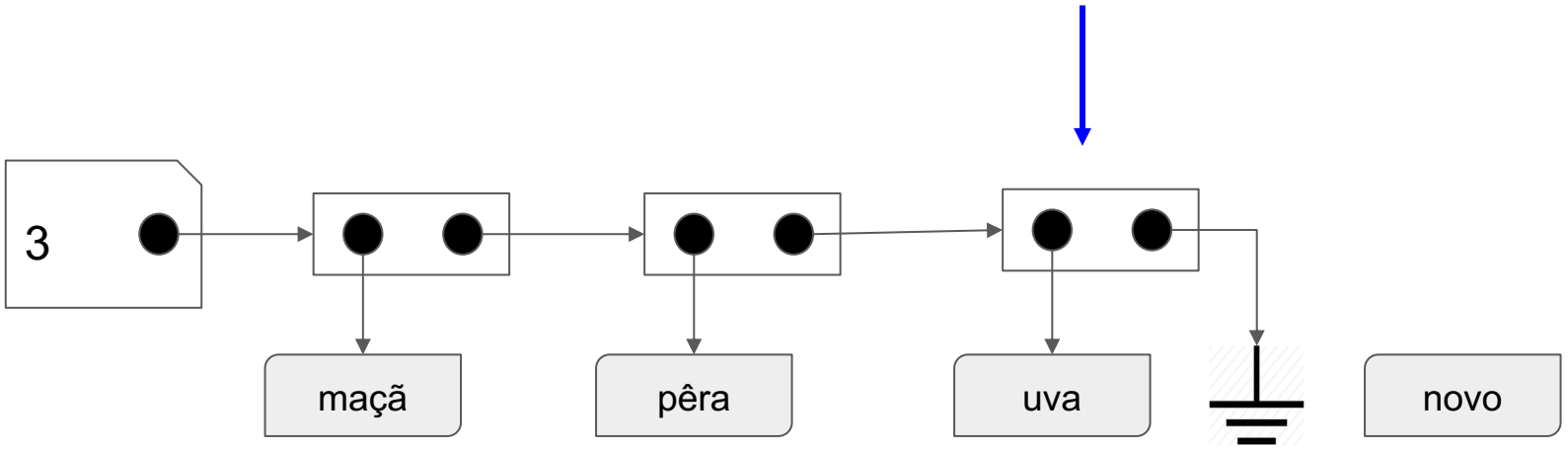
- É preciso modificar o último elemento da lista para fazer a inserção
- Para isso, é preciso primeiro percorrer a lista até esse elemento



Custo computacional: ??

Inserir um dado no fim da lista: detalhes

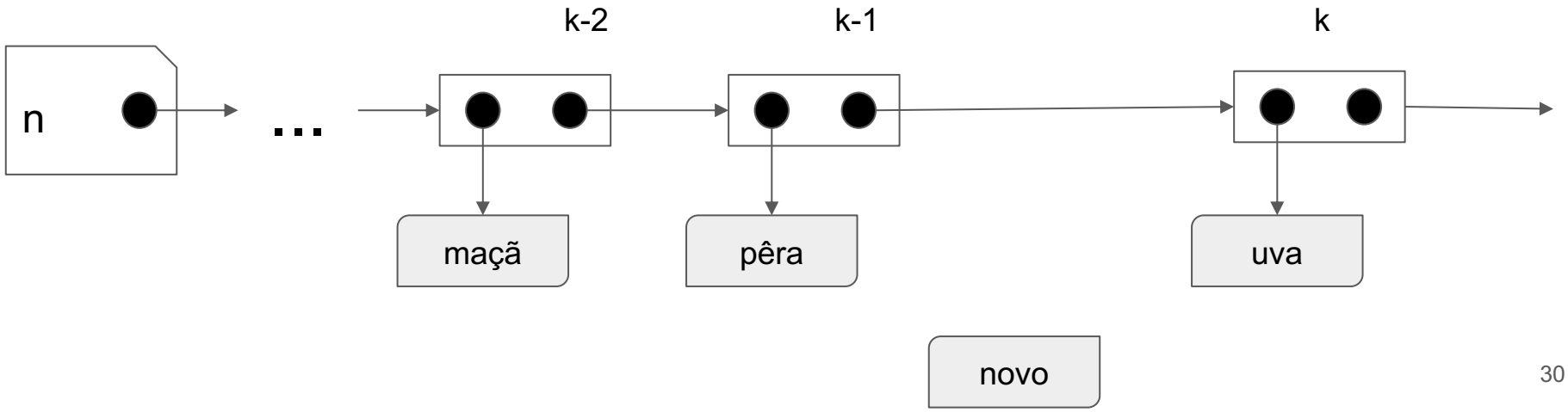
- É preciso modificar o último elemento da lista para fazer a inserção
- Para isso, é preciso primeiro percorrer a lista até esse elemento



Custo computacional: O(n)

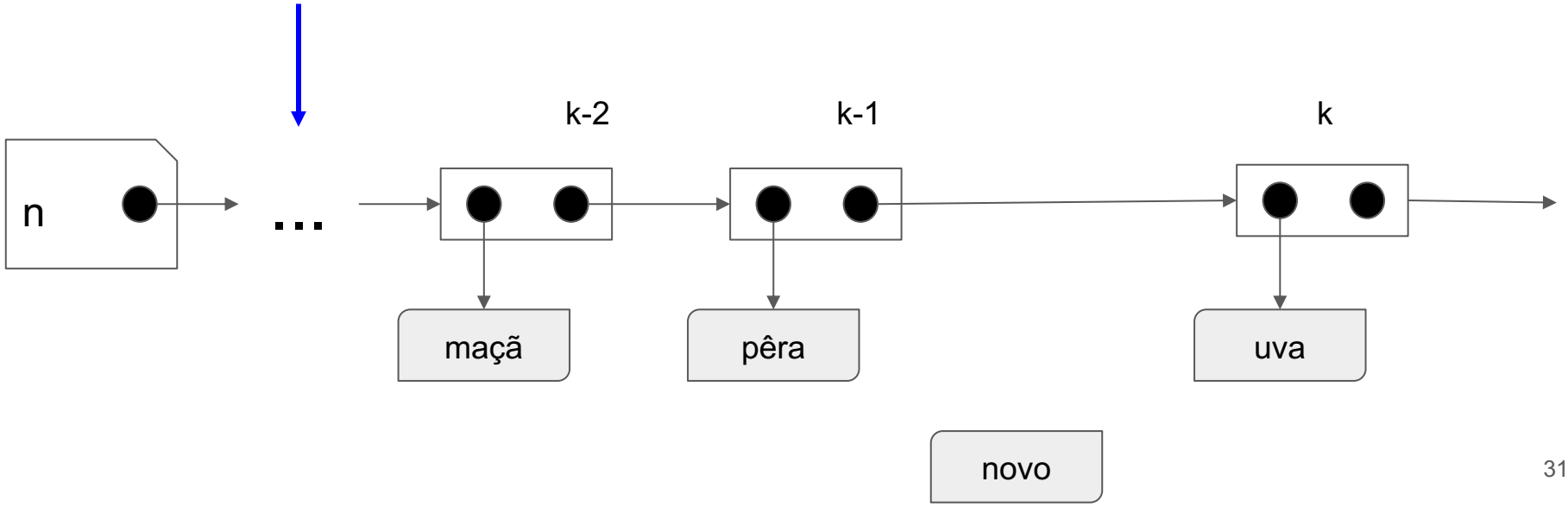
Inserir um dado na k-ésima posição, k>1

- Acessar elemento k-1
- Alocar novo elemento
- O novo elemento deve apontar para o dado e para próximo de k-1 (talvez nulo)
- O elemento k-1 deve apontar para o novo elemento
- Atualizar tamanho da lista



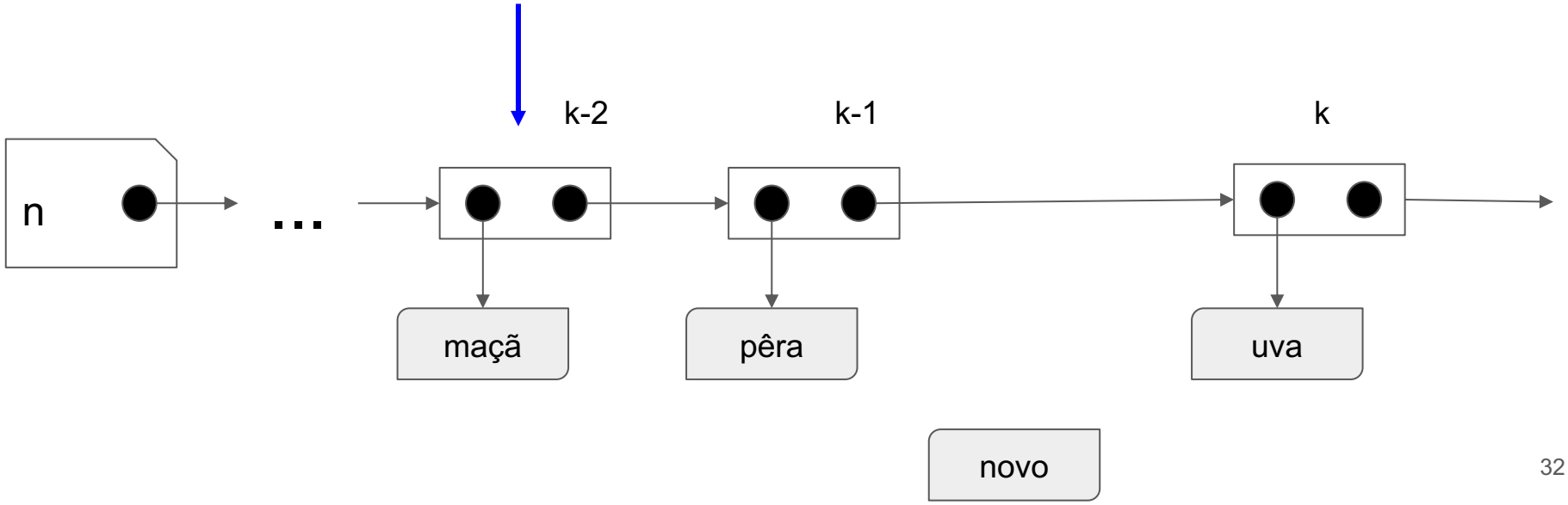
Inserir um dado na k-ésima posição, k>1

- Acessar elemento k-1
- Alocar novo elemento
- O novo elemento deve apontar para o dado e para próximo de k-1 (talvez nulo)
- O elemento k-1 deve apontar para o novo elemento
- Atualizar tamanho da lista



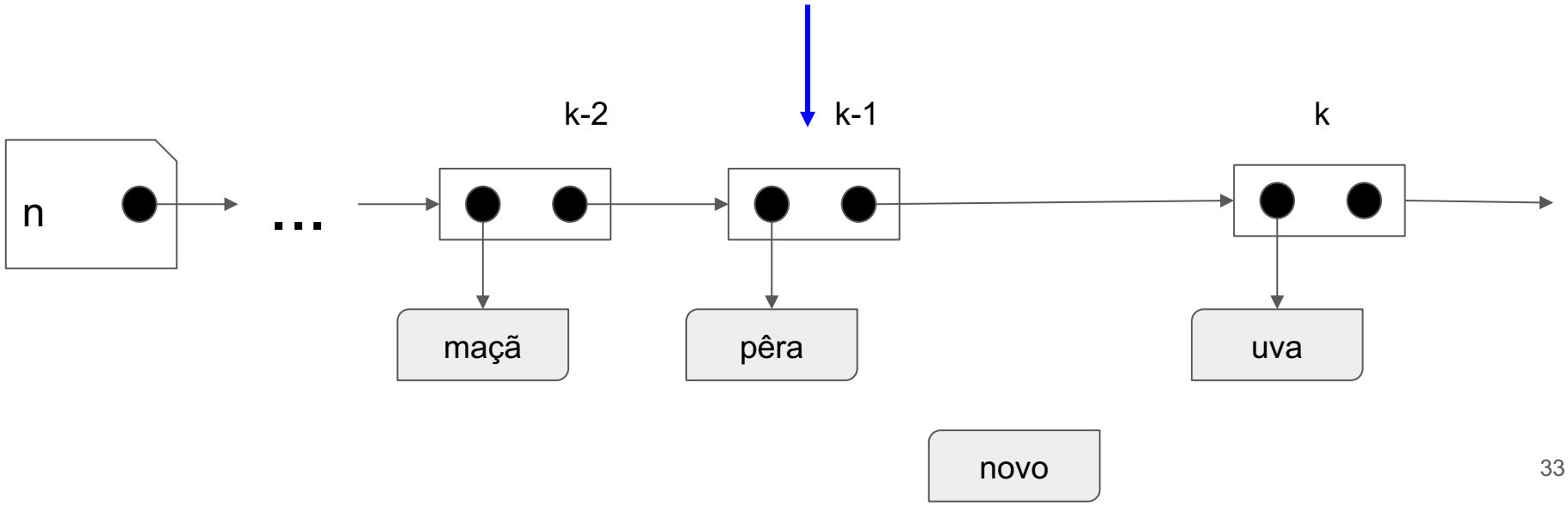
Inserir um dado na k-ésima posição, k>1

- Acessar elemento k-1
- Alocar novo elemento
- O novo elemento deve apontar para o dado e para próximo de k-1 (talvez nulo)
- O elemento k-1 deve apontar para o novo elemento
- Atualizar tamanho da lista



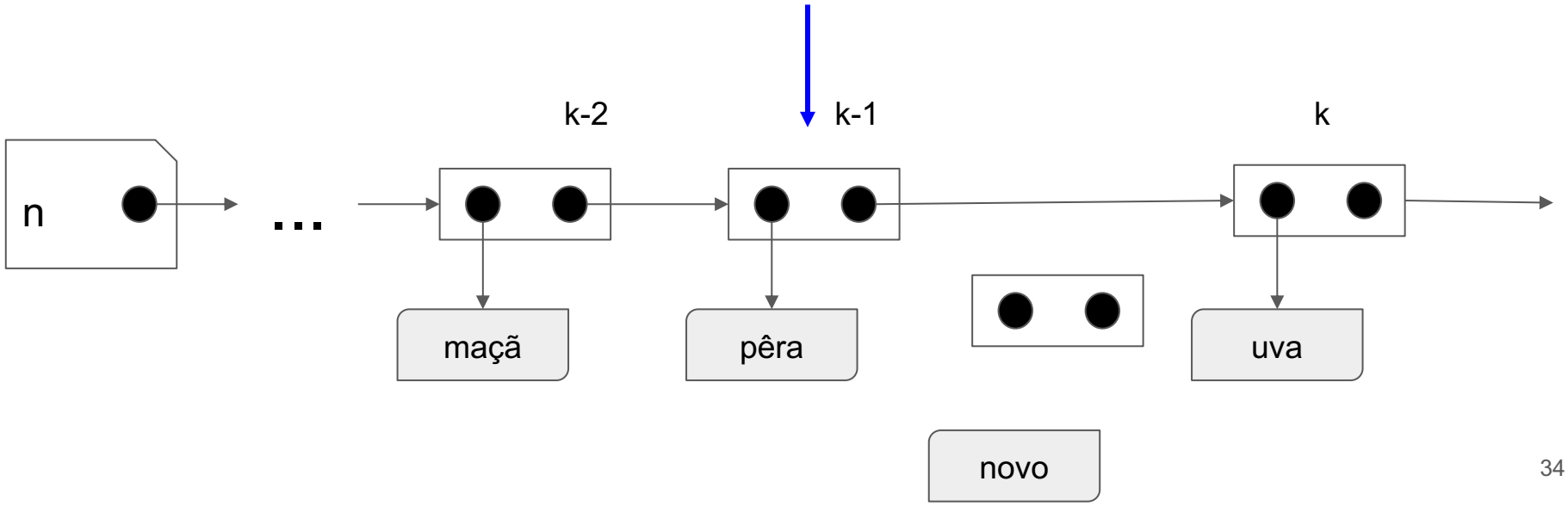
Inserir um dado na k-ésima posição, k>1

- Acessar elemento k-1
- Alocar novo elemento
- O novo elemento deve apontar para o dado e para próximo de k-1 (talvez nulo)
- O elemento k-1 deve apontar para o novo elemento
- Atualizar tamanho da lista



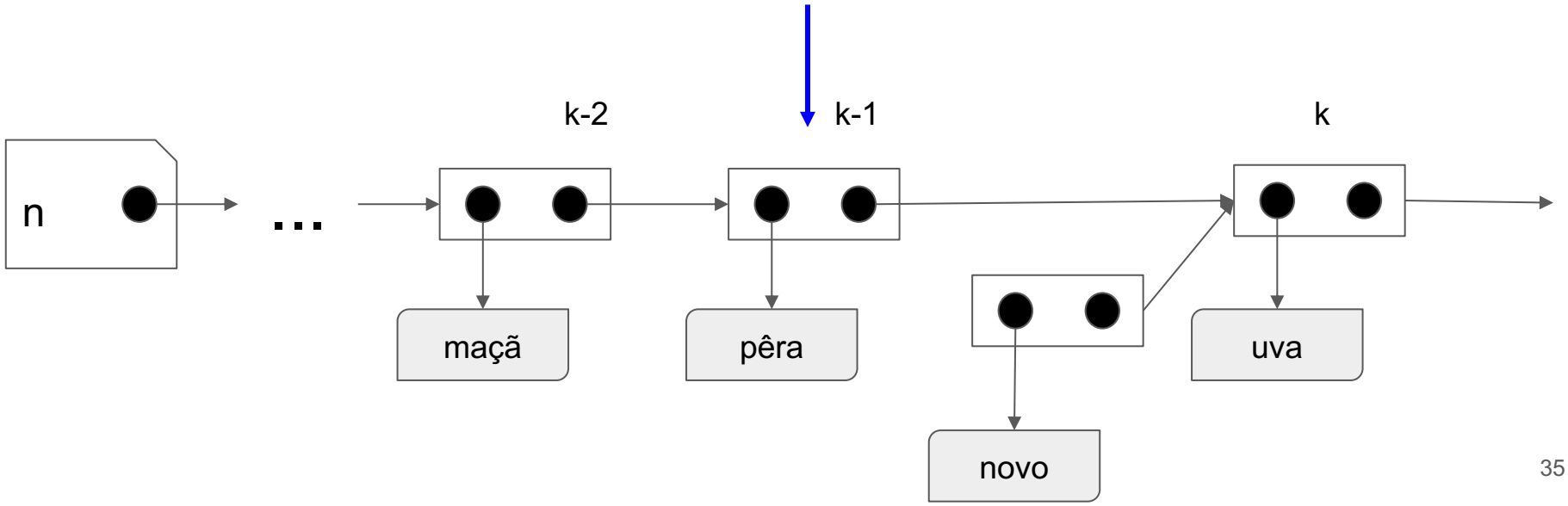
Inserir um dado na k-ésima posição, k>1

- Acessar elemento k-1
- Alocar novo elemento
- O novo elemento deve apontar para o dado e para próximo de k-1 (talvez nulo)
- O elemento k-1 deve apontar para o novo elemento
- Atualizar tamanho da lista



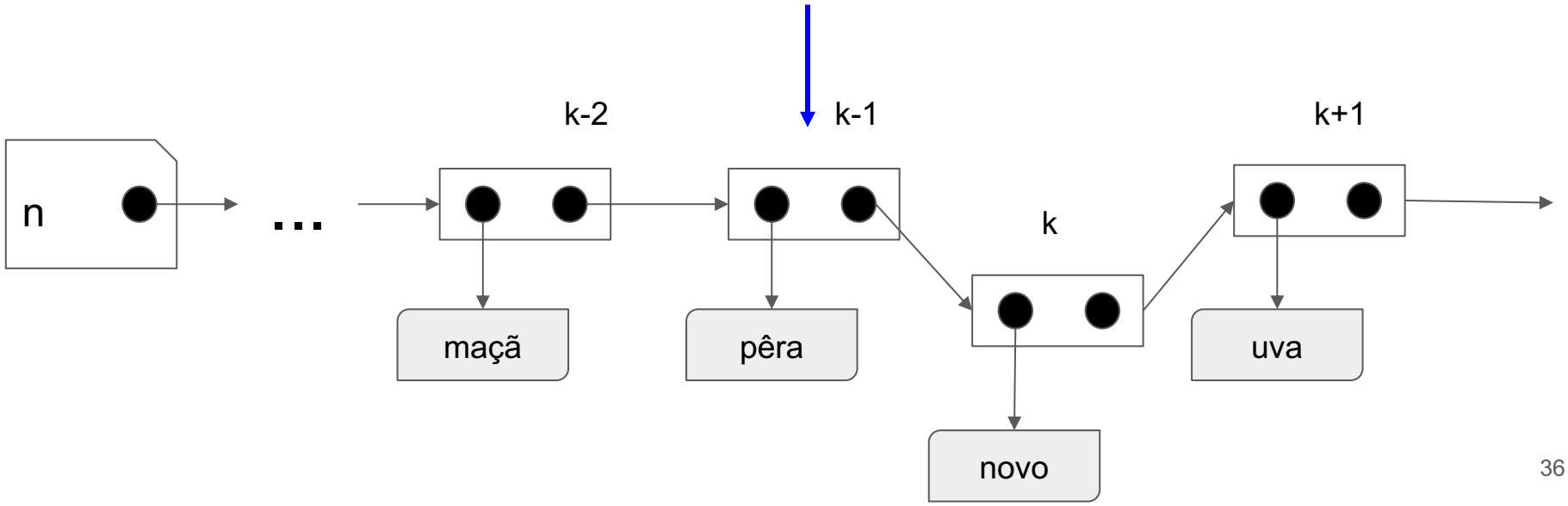
Inserir um dado na k-ésima posição, k>1

- Acessar elemento k-1
- Alocar novo elemento
- O novo elemento deve apontar para o dado e para próximo de k-1 (talvez nulo)
- O elemento k-1 deve apontar para o novo elemento
- Atualizar tamanho da lista



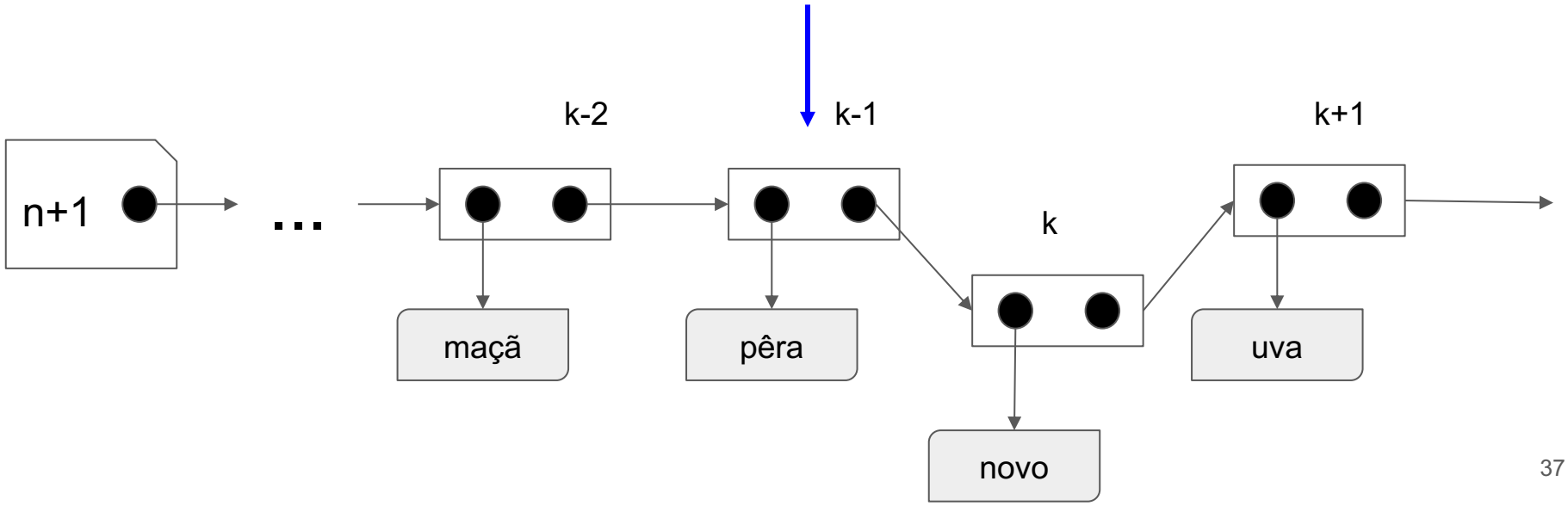
Inserir um dado na k-ésima posição, k>1

- Acessar elemento k-1
- Alocar novo elemento
- O novo elemento deve apontar para o dado e para próximo de k-1 (talvez nulo)
- O elemento k-1 deve apontar para o novo elemento
- Atualizar tamanho da lista



Inserir um dado na k-ésima posição, k>1

- Acessar elemento k-1
- Alocar novo elemento
- O novo elemento deve apontar para o dado e para próximo de k-1 (talvez nulo)
- O elemento k-1 deve apontar para o novo elemento
- Atualizar tamanho da lista



Generalizando a inserção em qualquer posição k>0

Inserir um dado na posição k de uma lista de tamanho N

- Aloque um novo elemento
- O novo elemento deve apontar para o dado a ser inserido
- Se k=1
  - O novo elemento deve apontar para o primeiro elemento da lista
  - A lista deve apontar para o novo elemento
- Senão se $1 < k \leq N+1$
  - Acesse o elemento k-1
  - O novo elemento deve apontar para o próximo de k-1 (viz, k talvez nulo)
  - O elemento k-1 deve apontar para o novo elemento
- Incrementar o tamanho da lista

Custo computacional no melhor caso?

Generalizando a inserção em qualquer posição k>0

Inserir um dado na posição k de uma lista de tamanho N

- Aloque um novo elemento
- O novo elemento deve apontar para o dado a ser inserido
- Se k=1
  - O novo elemento deve apontar para o primeiro elemento da lista
  - A lista deve apontar para o novo elemento
- Senão se $1 < k \leq N+1$
  - Acesse o elemento k-1
  - O novo elemento deve apontar para o próximo de k-1 (talvez nulo)
  - O elemento k-1 deve apontar para o novo elemento
- Incrementar o tamanho da lista

Custo computacional no melhor caso = O(1)

Generalizando a inserção em qualquer posição k>0

Inserir um dado na posição k de uma lista de tamanho N

- Aloque um novo elemento
- O novo elemento deve apontar para o dado a ser inserido
- Se k=1
  - O novo elemento deve apontar para o primeiro elemento da lista
  - A lista deve apontar para o novo elemento
- Senão se $1 < k \leq N+1$
  - Acesse o elemento k-1
  - O novo elemento deve apontar para o próximo de k-1 (talvez nulo)
  - O elemento k-1 deve apontar para o novo elemento
- Incrementar o tamanho da lista

Custo computacional no pior caso?

Generalizando a inserção em qualquer posição k>0

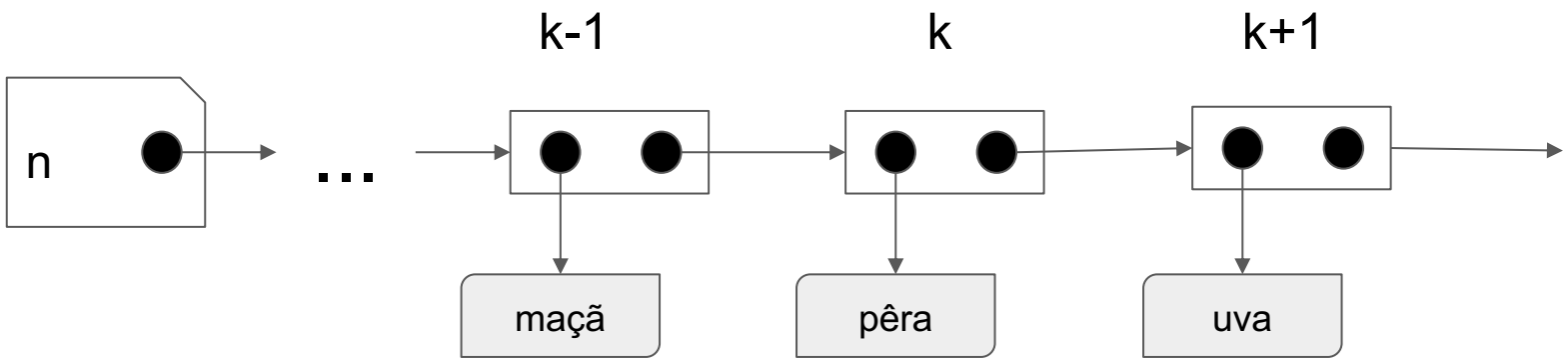
Inserir um dado na posição k de uma lista de tamanho N

- Aloque um novo elemento
- O novo elemento deve apontar para o dado a ser inserido
- Se k=1
  - O novo elemento deve apontar para o primeiro elemento da lista
  - A lista deve apontar para o novo elemento
- Senão se $1 < k \leq N+1$
  - Acesse o elemento k-1
  - O novo elemento deve apontar para o próximo de k-1 (talvez nulo)
  - O elemento k-1 deve apontar para o novo elemento
- Incrementar o tamanho da lista

Custo computacional no melhor caso = O(n)

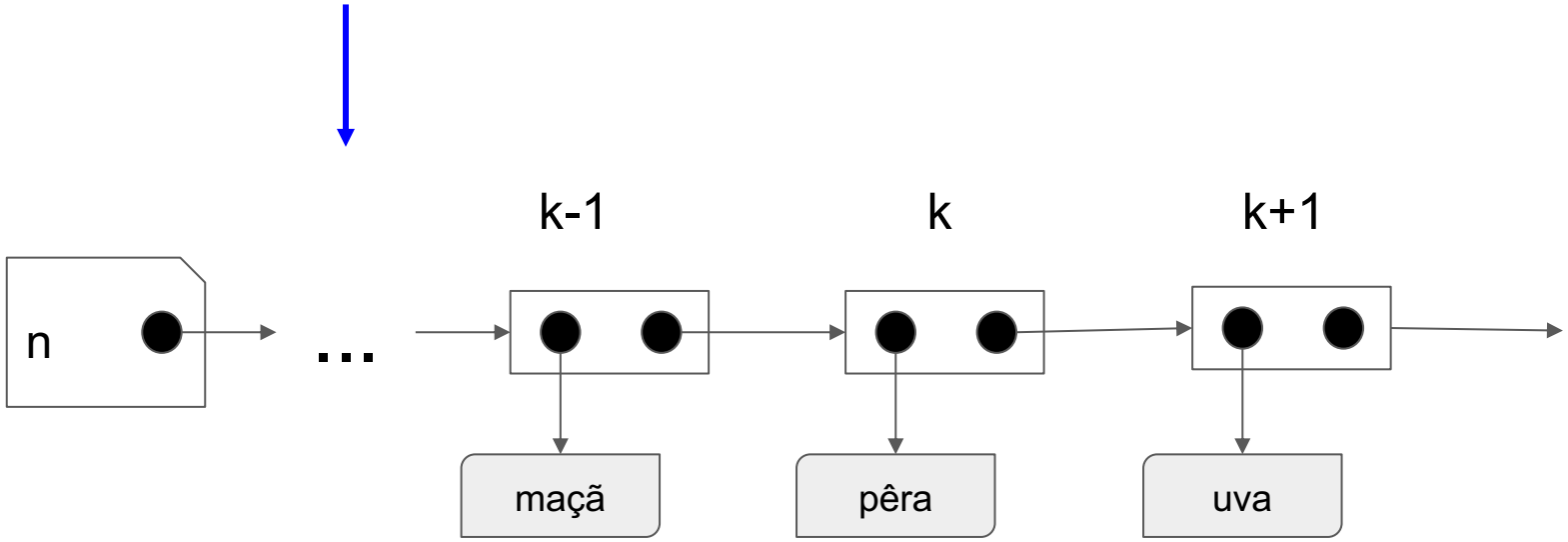
Obter k-ésimo dado da lista

- Acessar elemento k
- Retornar o dado apontado pelo elemento k



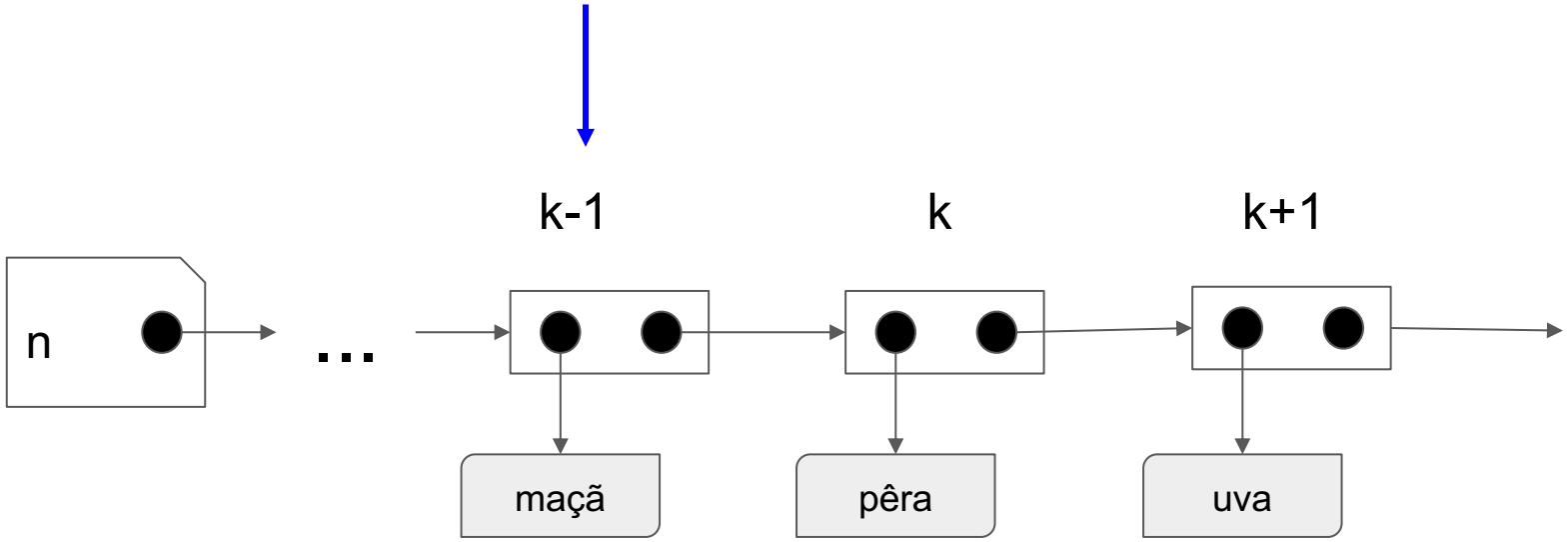
Obter k-ésimo dado da lista

- Acessar elemento k
- Retornar o dado apontado pelo elemento k



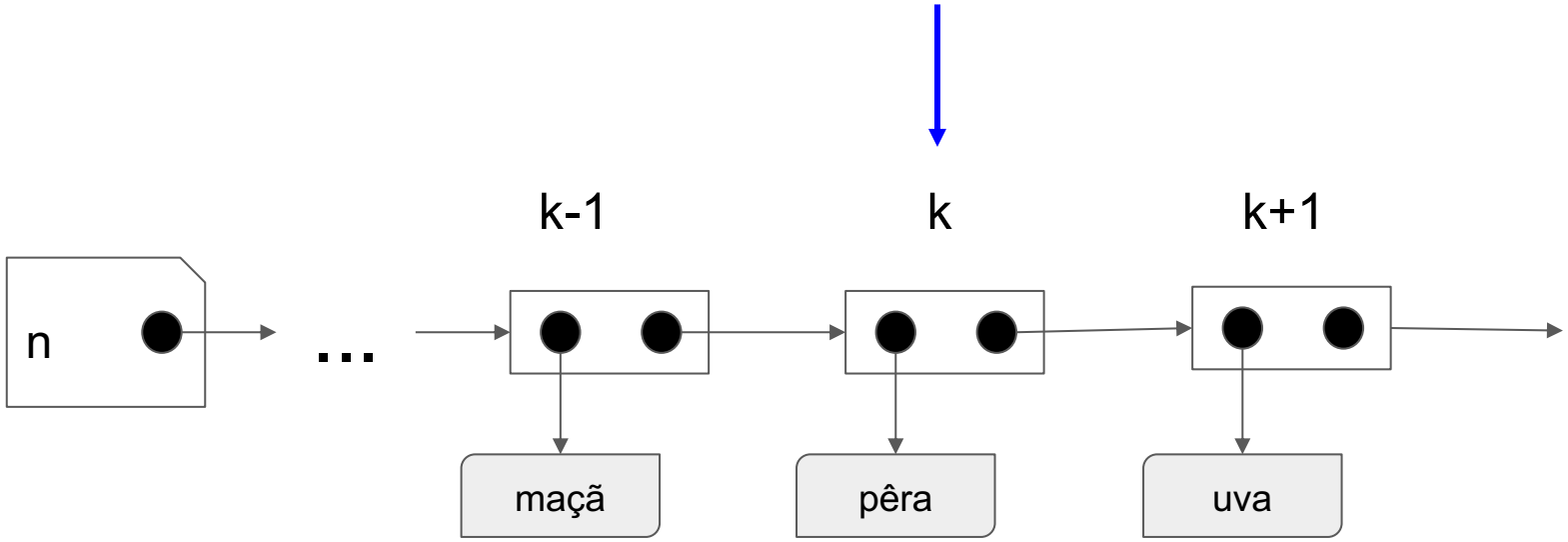
Obter k-ésimo dado da lista

- Acessar elemento k
- Retornar o dado apontado pelo elemento k



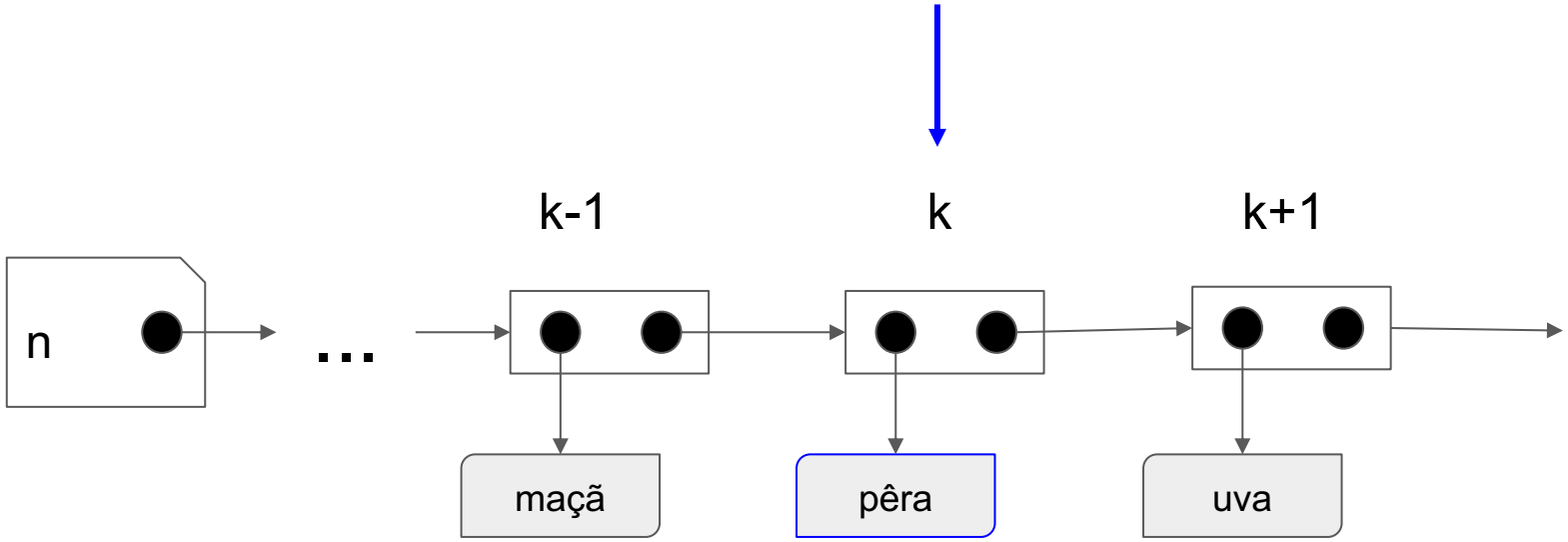
Obter k-ésimo dado da lista

- Acessar elemento k
- Retornar o dado apontado pelo elemento k



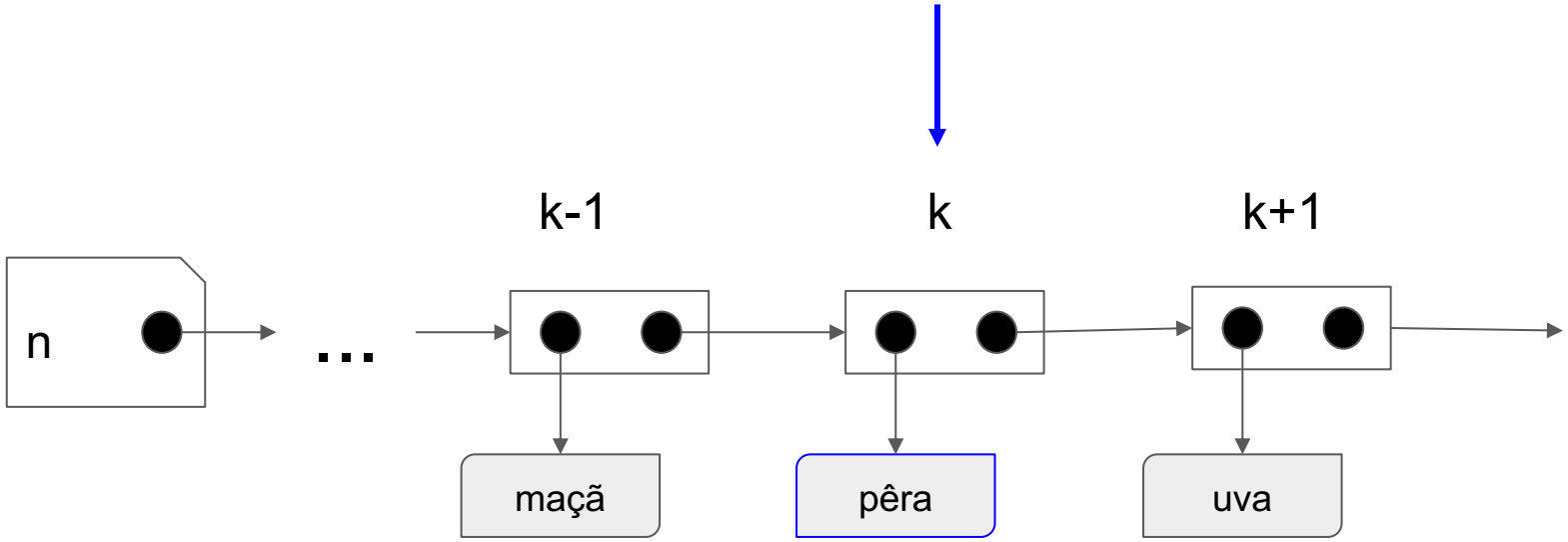
Obter k-ésimo dado da lista

- Acessar elemento k
- Retornar o dado apontado pelo elemento k



Obter k-ésimo dado da lista

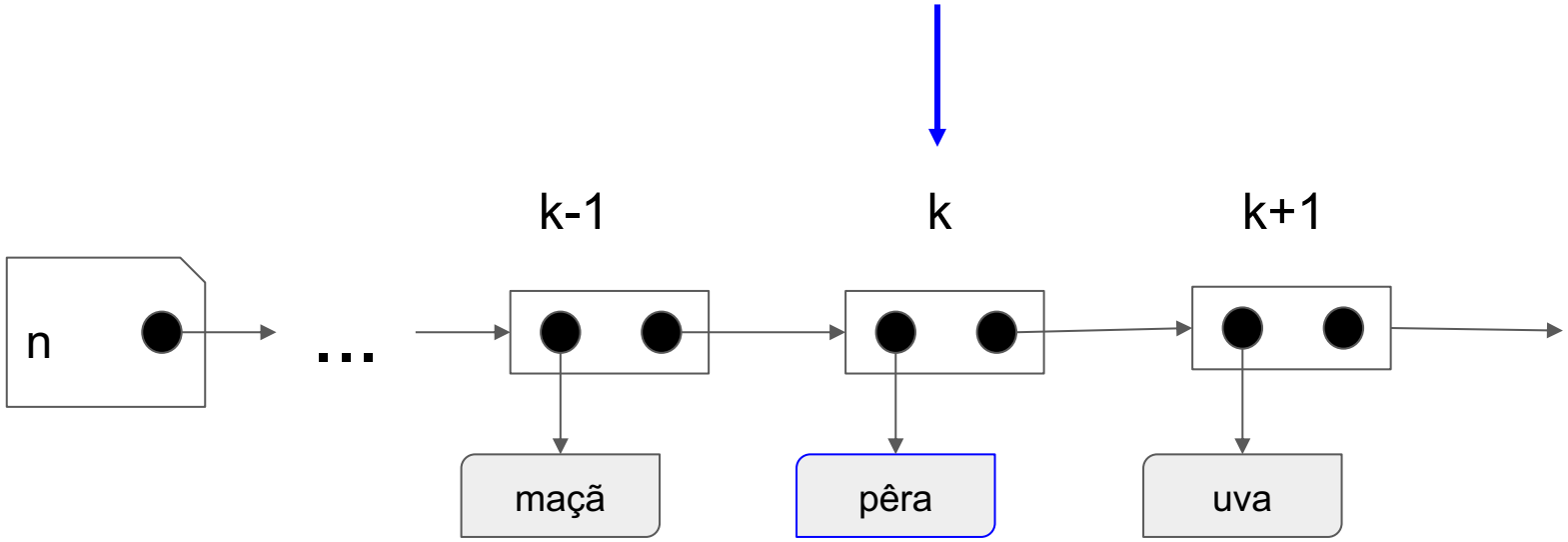
- Acessar elemento k
- Retornar o dado apontado pelo elemento k



Custo computacional no melhor caso = O(1)

Obter k-ésimo dado da lista

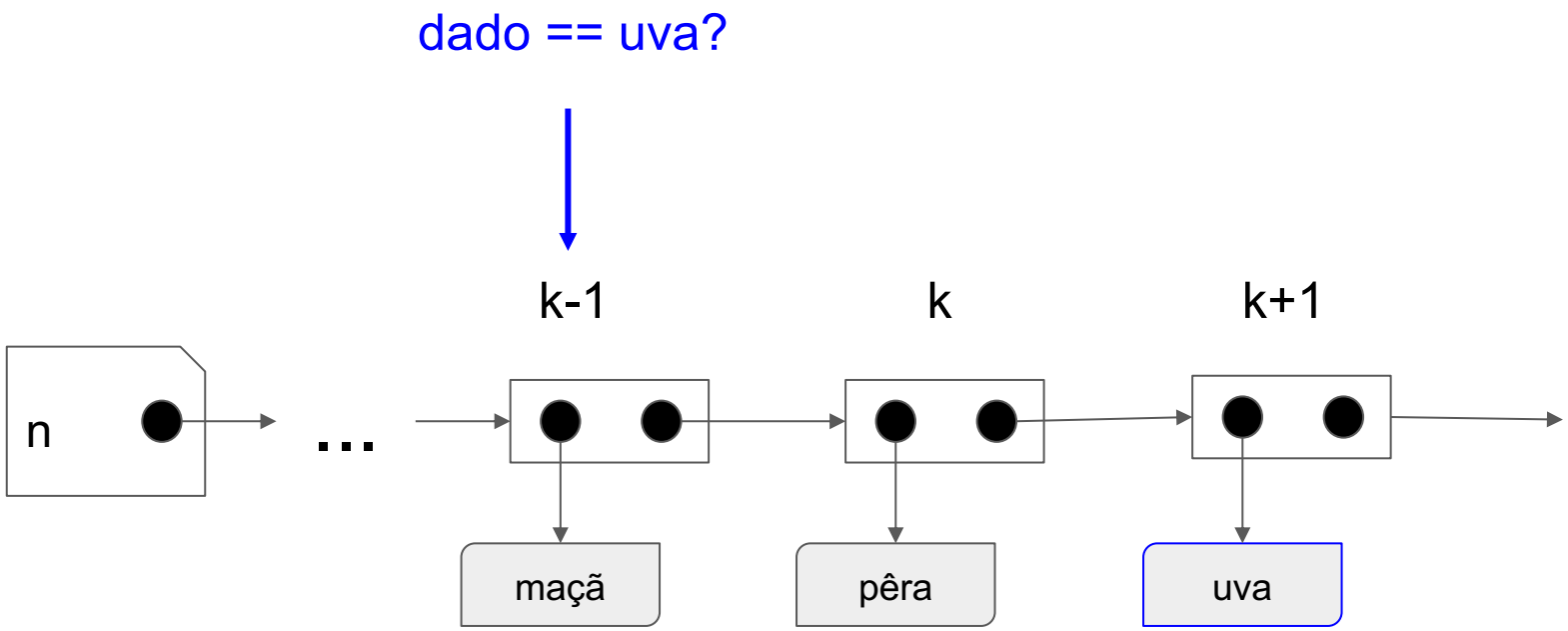
- Acessar elemento k
- Retornar o dado apontado pelo elemento k



Custo computacional no pior caso = O(n)

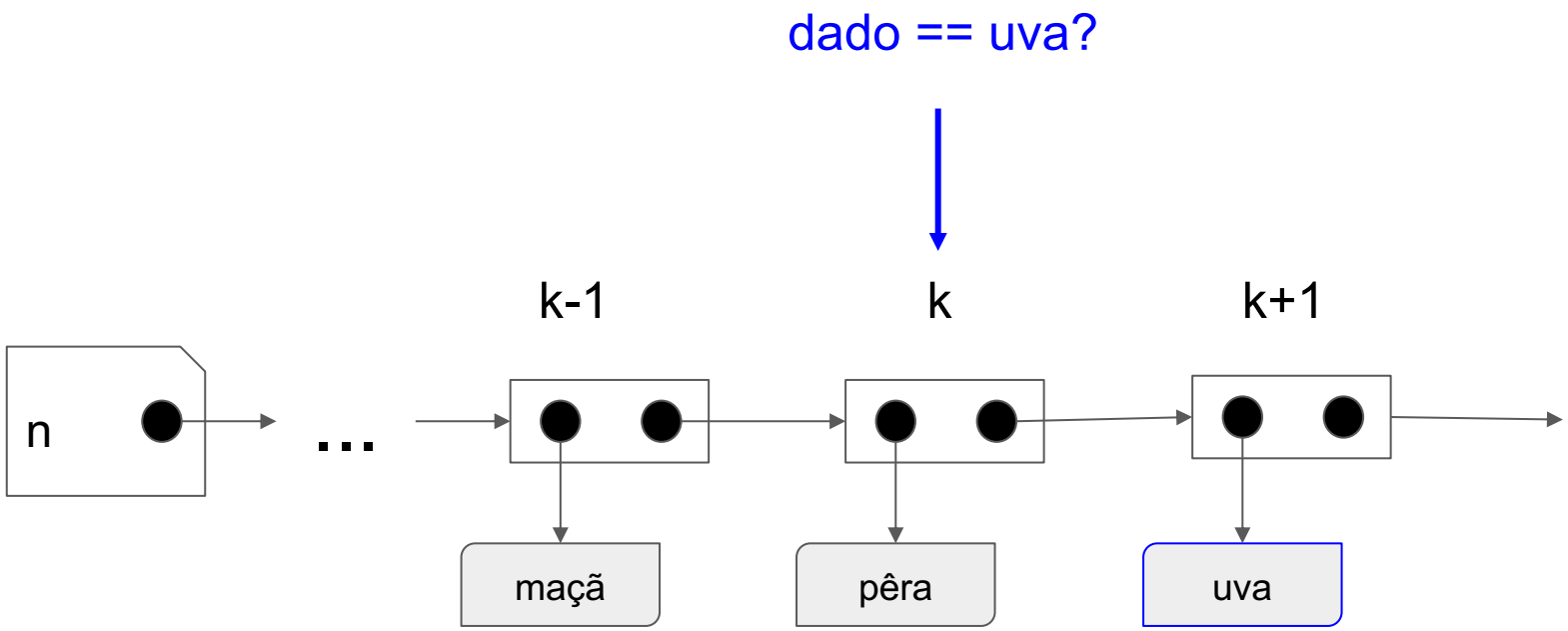
Buscar um dado na lista

1. Visitar elemento a elemento até encontrar o dado ou fim da lista
2. Retornar índice k do elemento que contém o dado buscado ou -1



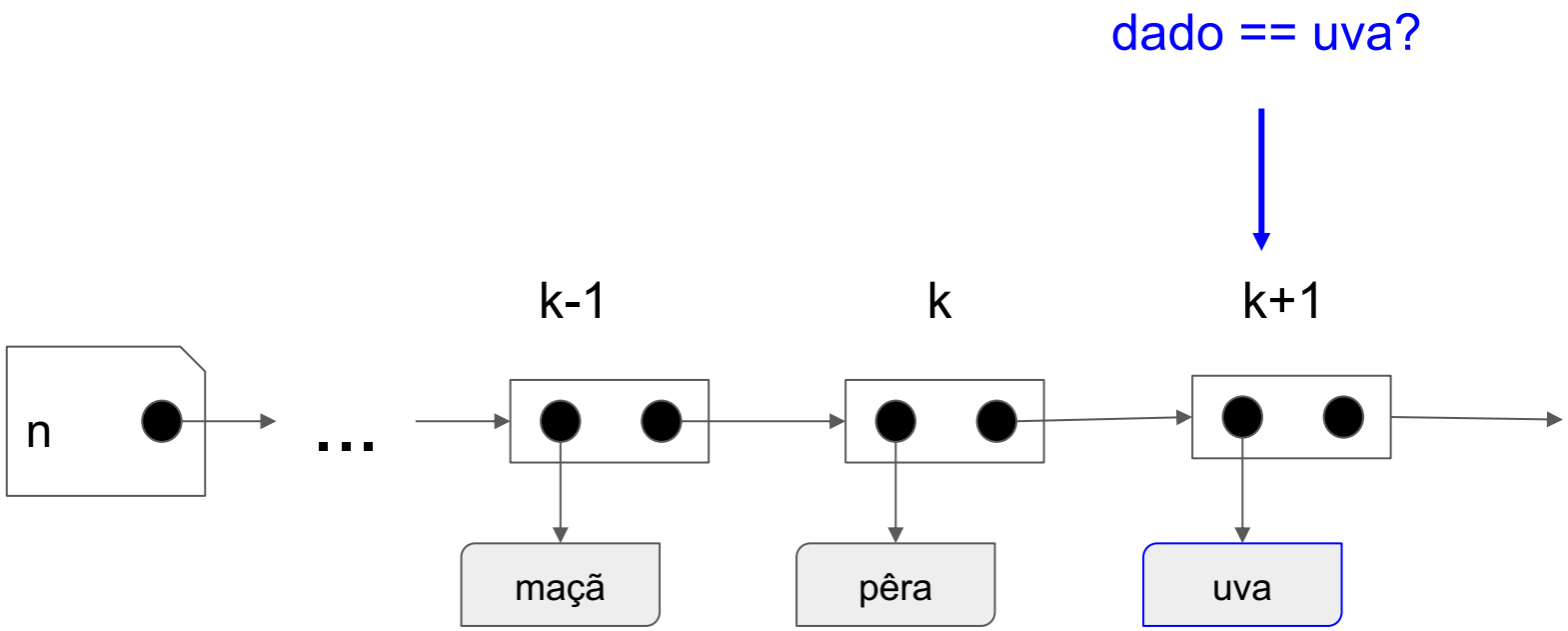
Buscar um dado na lista

1. Visitar elemento a elemento até encontrar o dado ou fim da lista
2. Retornar índice k do elemento que contém o dado buscado ou -1



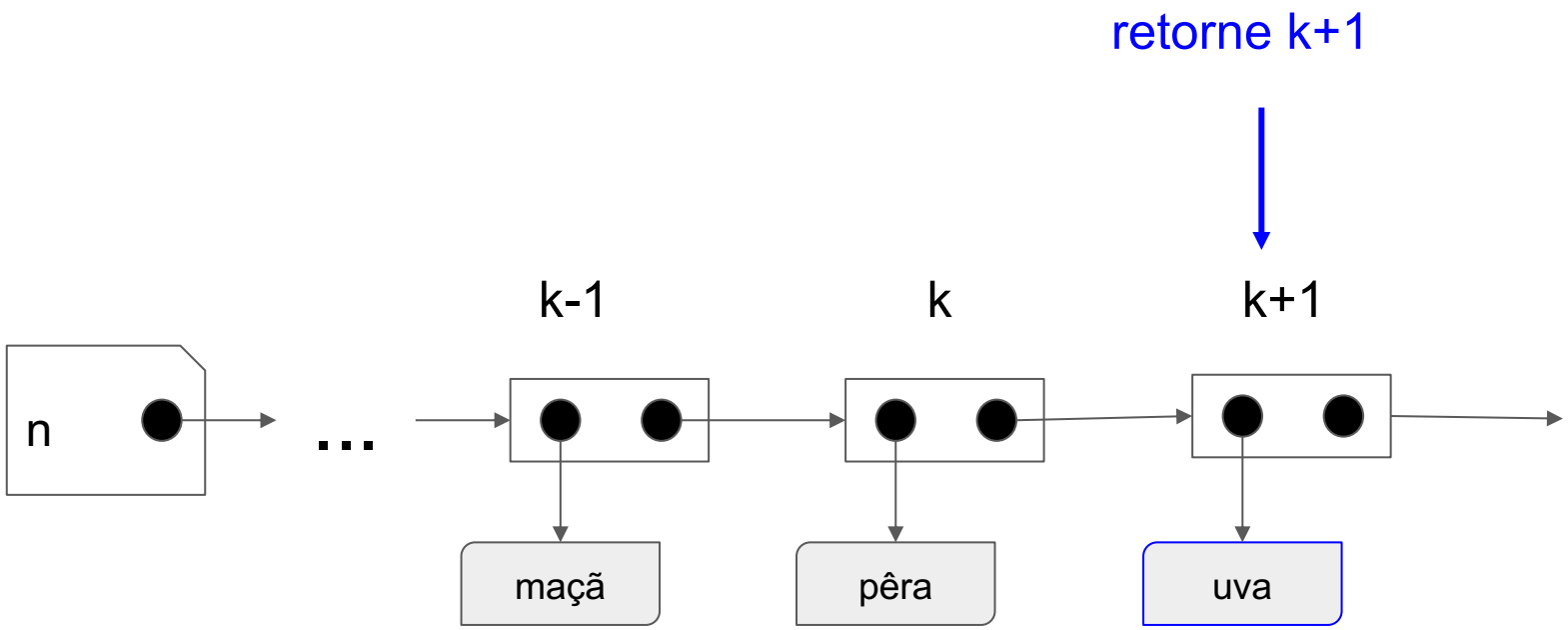
Buscar um dado na lista

1. Visitar elemento a elemento até encontrar o dado ou fim da lista
2. Retornar índice k do elemento que contém o dado buscado ou -1



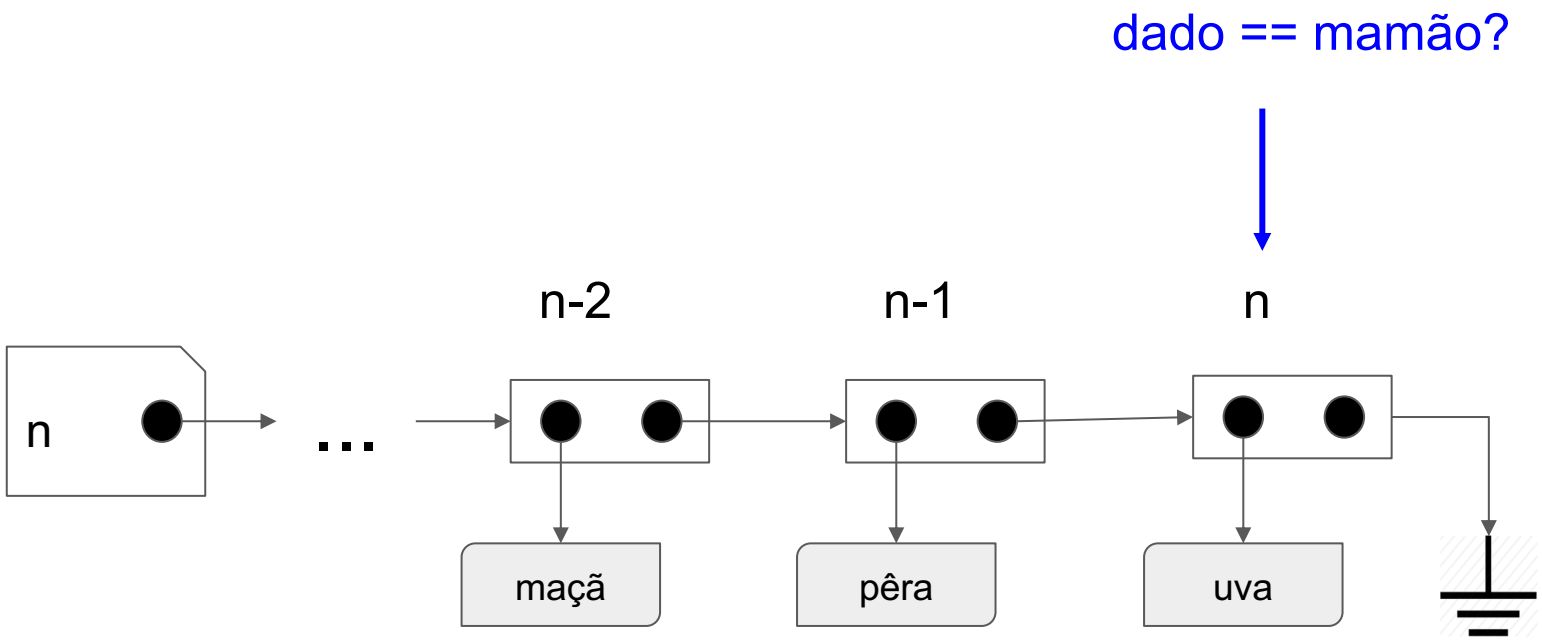
Buscar um dado na lista

1. Visitar elemento a elemento até encontrar o dado ou fim da lista
2. Retornar índice k do elemento que contém o dado buscado ou -1



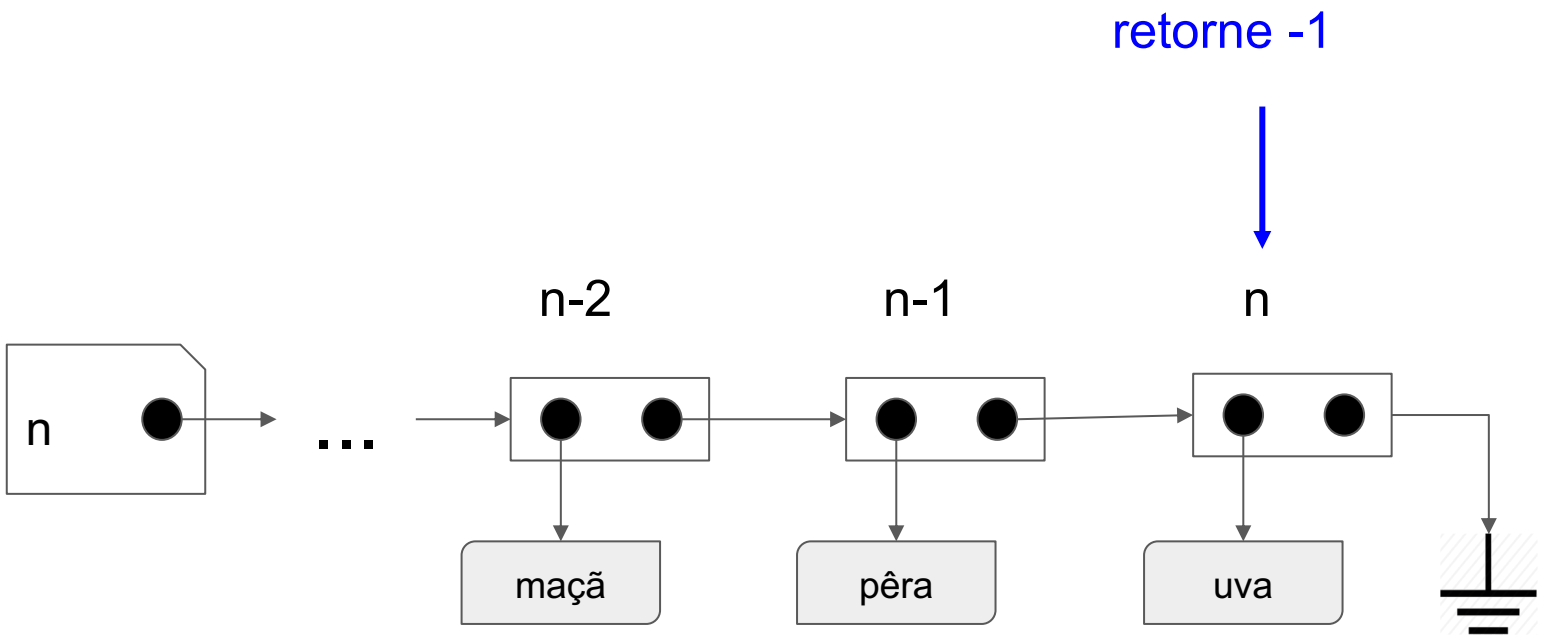
Buscar um dado na lista

1. Visitar elemento a elemento até encontrar o dado ou fim da lista
2. Retornar índice k do elemento que contém o dado buscado ou -1



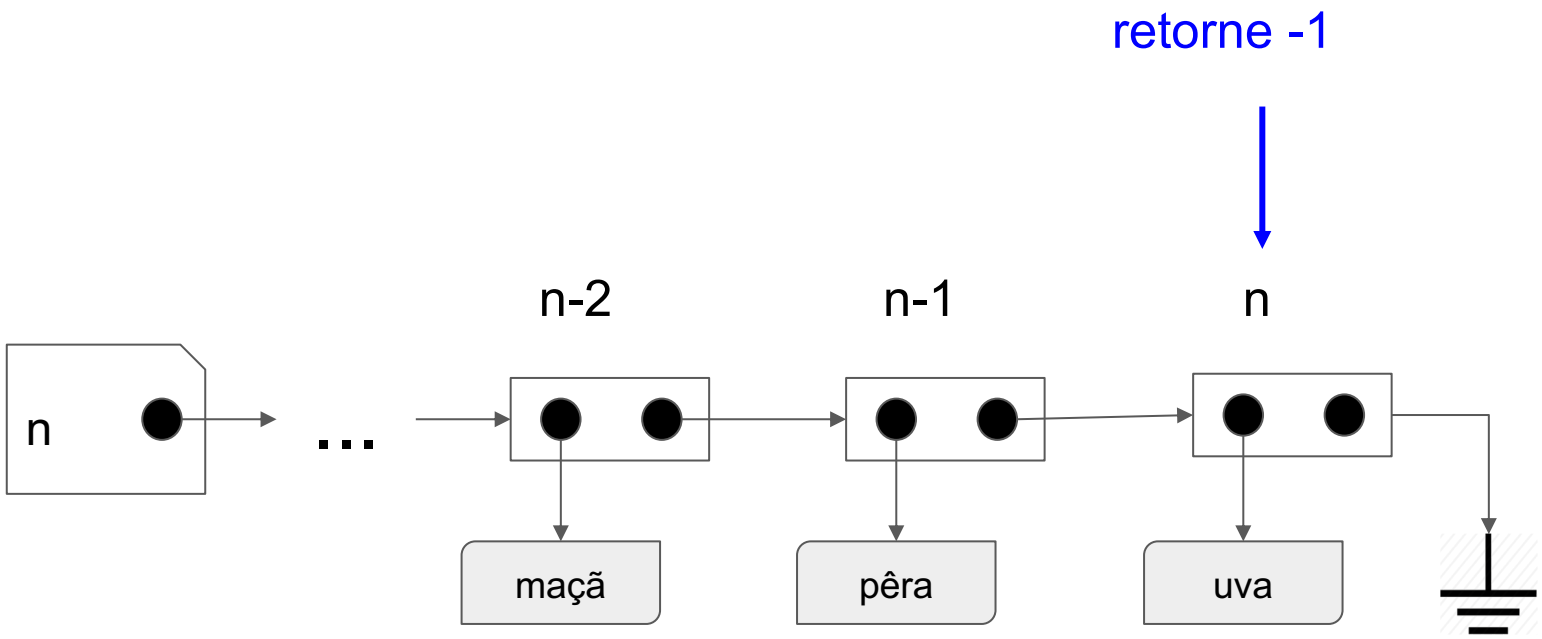
Buscar um dado na lista

1. Visitar elemento a elemento até encontrar o dado ou fim da lista
2. Retornar índice k do elemento que contém o dado buscado ou -1



Buscar um dado na lista

1. Visitar elemento a elemento até encontrar o dado ou fim da lista
2. Retornar índice k do elemento que contém o dado buscado ou -1



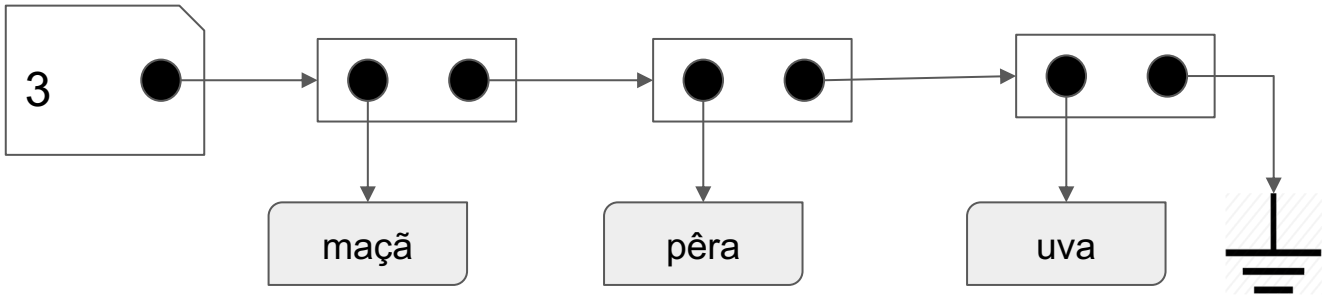
Custo computacional no pior caso = O(n)

Remover k-ésimo dado da lista

- Como na inserção, distinguir entre os dois casos
  - k=1: remover o primeiro dado da lista
  - k>1: remover um dado entre a segunda e última posição
- Porém agora:
  - Precisamos desalocar o elemento que contém o dado a ser removido
  - Precisamos retornar o dado removido
  - São necessárias variáveis auxiliares para não perder referências

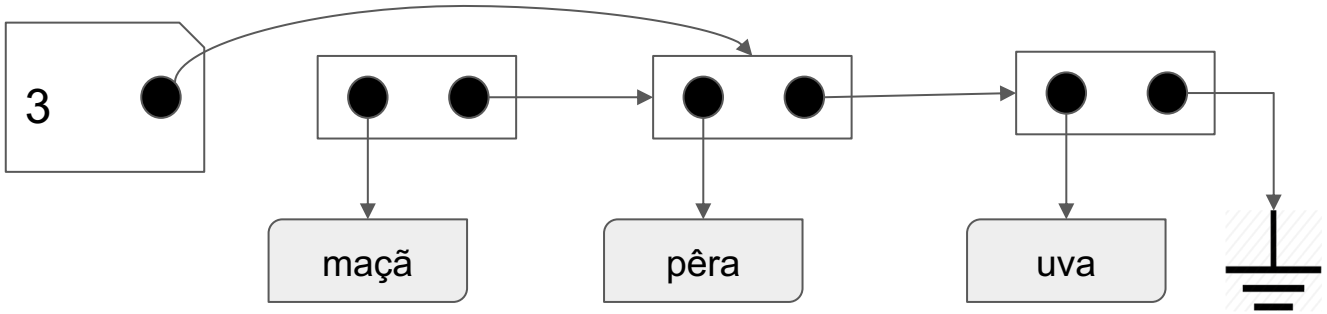
Remover o primeiro dado da lista: passos

1. A lista deve apontar para o próximo do primeiro elemento
2. Desalocar o primeiro elemento
3. Decrementar o tamanho da lista
4. Retornar o dado que estava no primeiro elemento



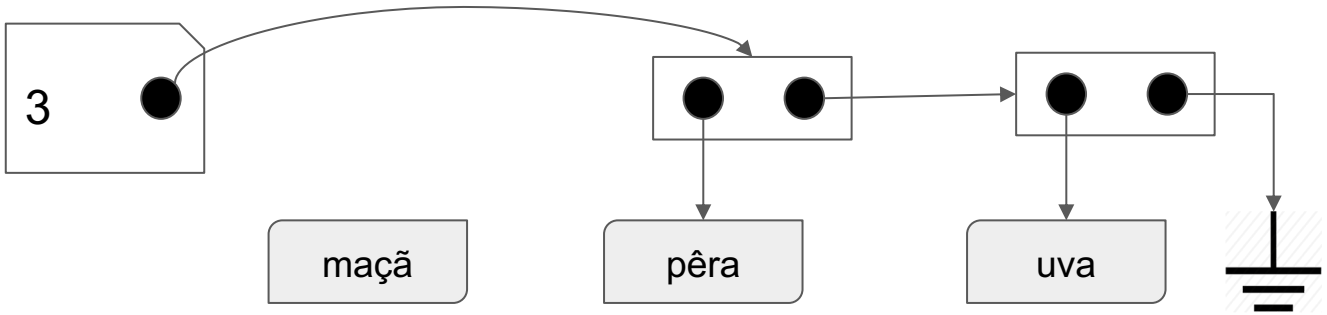
Remover o primeiro dado da lista: passos

1. A lista deve apontar para o próximo do primeiro elemento
2. Desalocar o primeiro elemento
3. Decrementar o tamanho da lista
4. Retornar o dado que estava no primeiro elemento



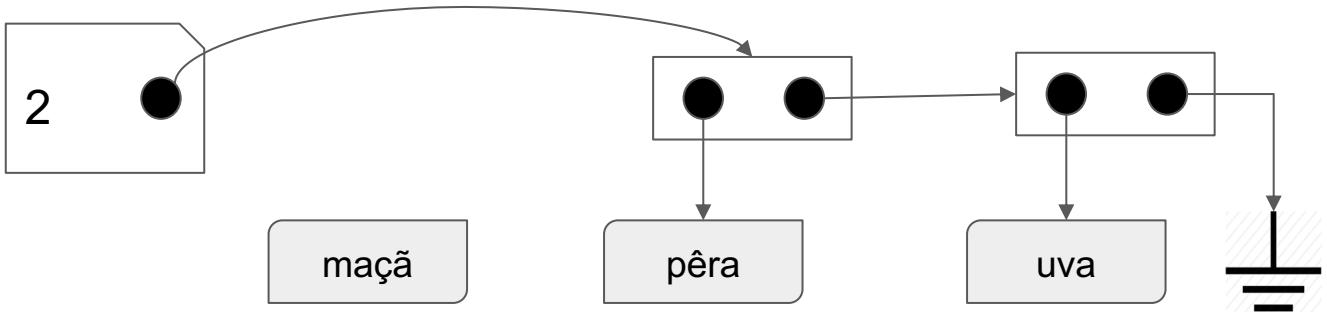
Remover o primeiro dado da lista: passos

1. A lista deve apontar para o próximo do primeiro elemento
2. Desalocar o primeiro elemento
3. Decrementar o tamanho da lista
4. Retornar o dado que estava no primeiro elemento



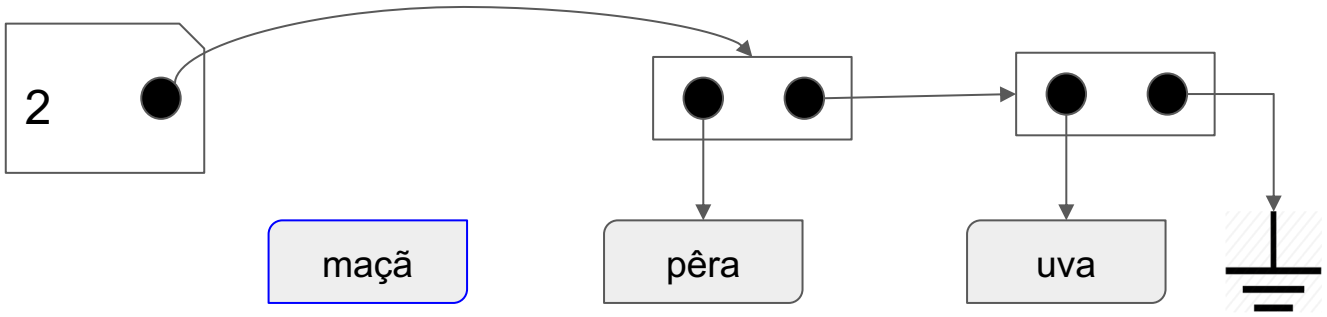
Remover o primeiro dado da lista: passos

1. A lista deve apontar para o próximo do primeiro elemento
2. Desalocar o primeiro elemento
3. Decrementar o tamanho da lista
4. Retornar o dado que estava no primeiro elemento



Remover o primeiro dado da lista: passos

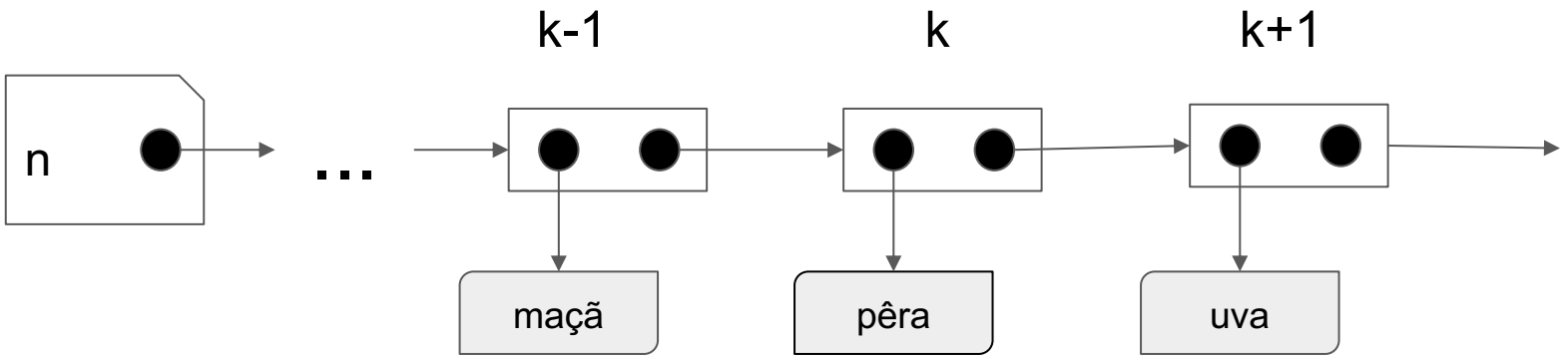
1. A lista deve apontar para o próximo do primeiro elemento
2. Desalocar o primeiro elemento
3. Decrementar o tamanho da lista
4. Retornar o dado que estava no primeiro elemento



Custo computacional = O(1)

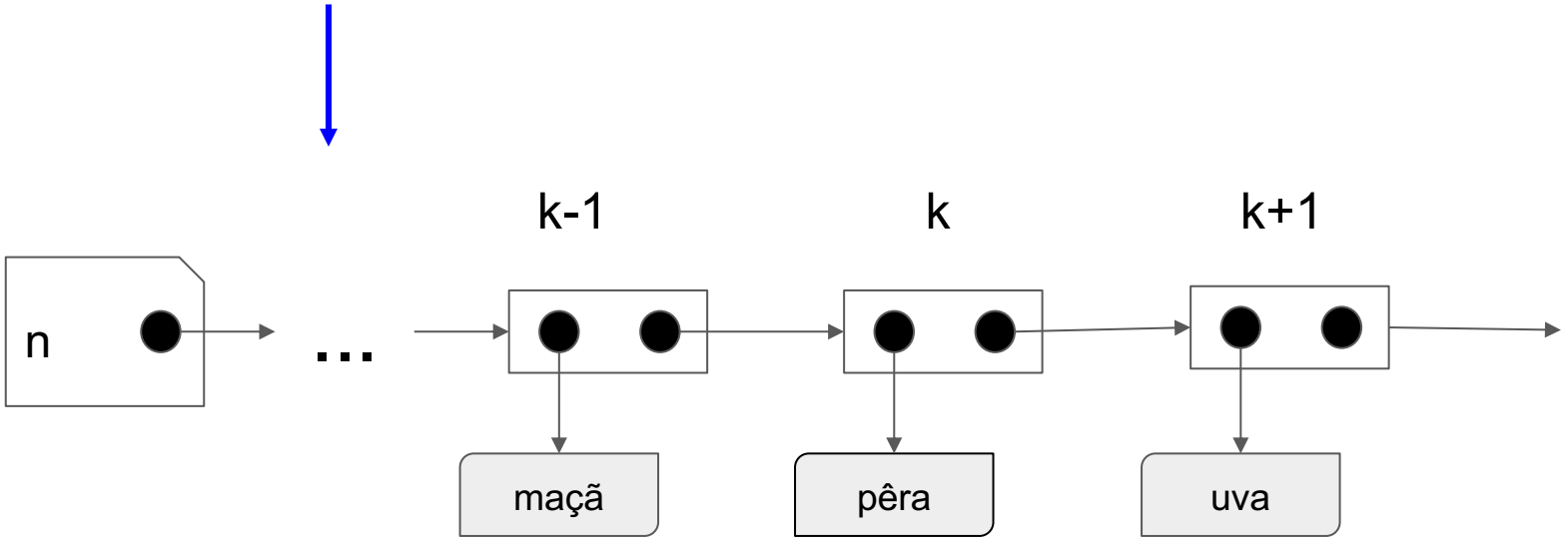
Remover o k-ésimo dado da lista, k>1

1. Acesse o elemento k-1
2. O elemento k-1 deve apontar para o próximo do elemento k (talvez nulo)
3. Desalocar o elemento k
4. Decrementar o tamanho da lista
5. Retornar o dado que estava no elemento k



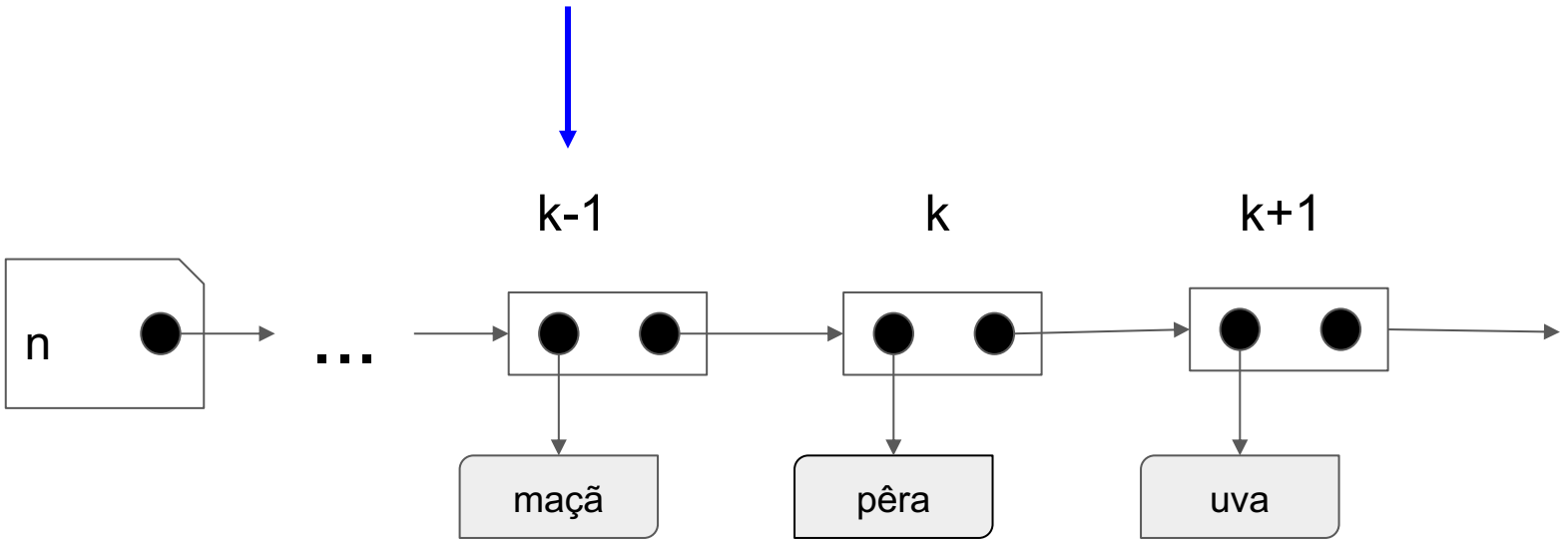
Remover o k-ésimo dado da lista, k>1

1. Acesse o elemento k-1
2. O elemento k-1 deve apontar para o próximo do elemento k (talvez nulo)
3. Desalocar o elemento k
4. Decrementar o tamanho da lista
5. Retornar o dado que estava no elemento k



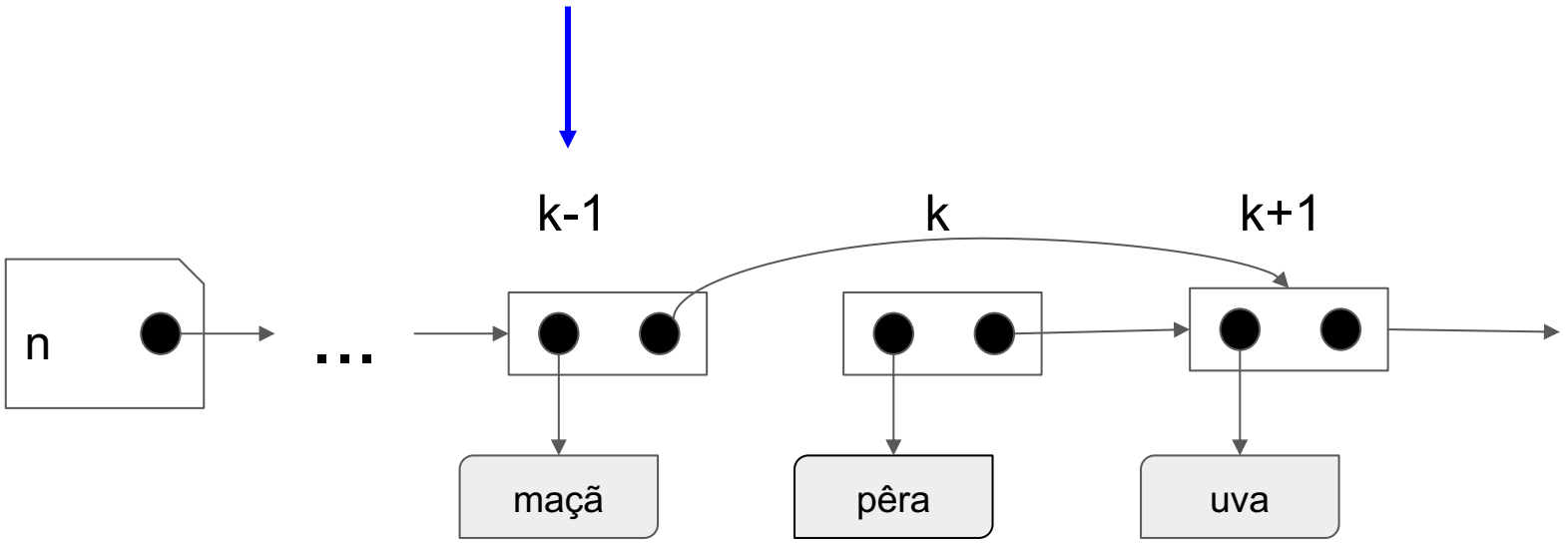
Remover o k-ésimo dado da lista, k>1

1. Acesse o elemento k-1
2. O elemento k-1 deve apontar para o próximo do elemento k (talvez nulo)
3. Desalocar o elemento k
4. Decrementar o tamanho da lista
5. Retornar o dado que estava no elemento k



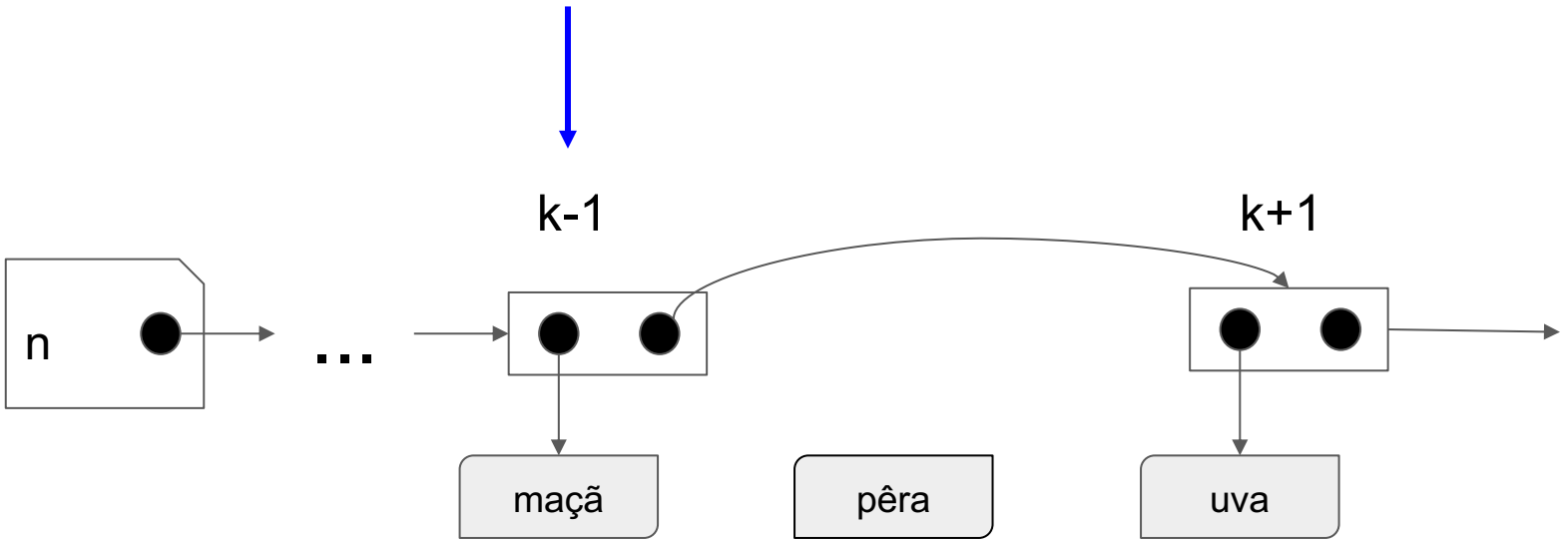
Remover o k-ésimo dado da lista, k>1

1. Acesse o elemento k-1
2. O elemento k-1 deve apontar para o próximo do elemento k (talvez nulo)
3. Desalocar o elemento k
4. Decrementar o tamanho da lista
5. Retornar o dado que estava no elemento k



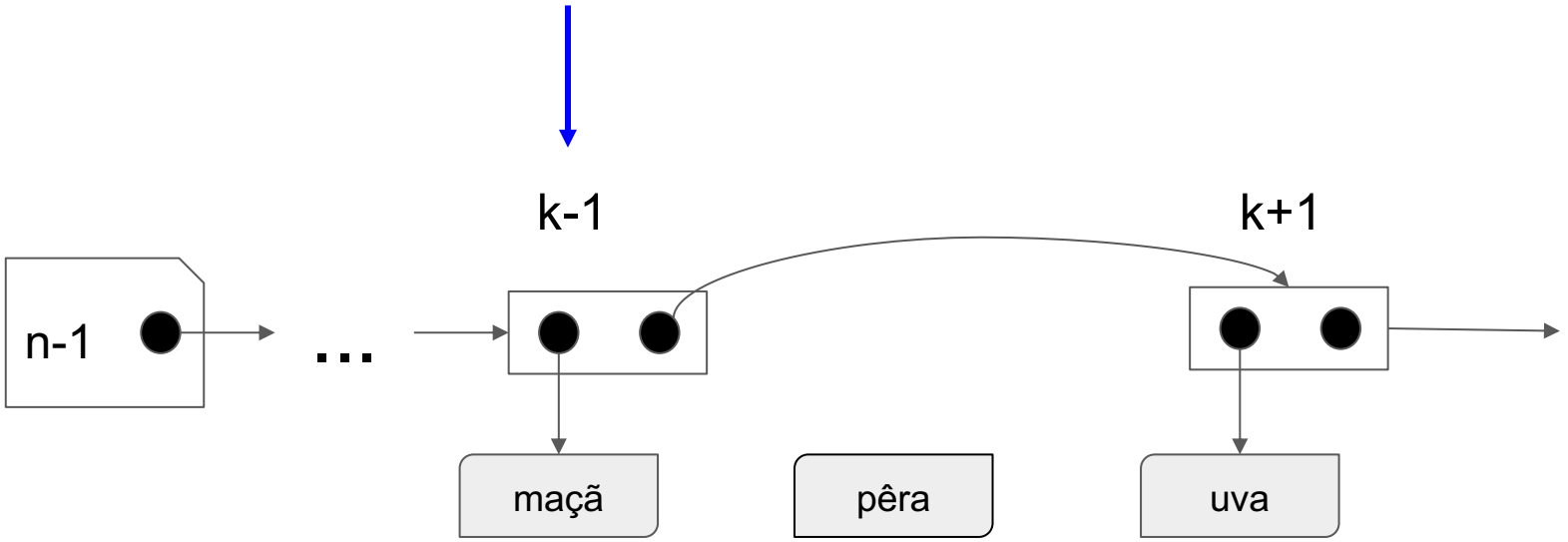
Remover o k-ésimo dado da lista, k>1

1. Acesse o elemento k-1
2. O elemento k-1 deve apontar para o próximo do elemento k (talvez nulo)
3. Desalocar o elemento k
4. Decrementar o tamanho da lista
5. Retornar o dado que estava no elemento k



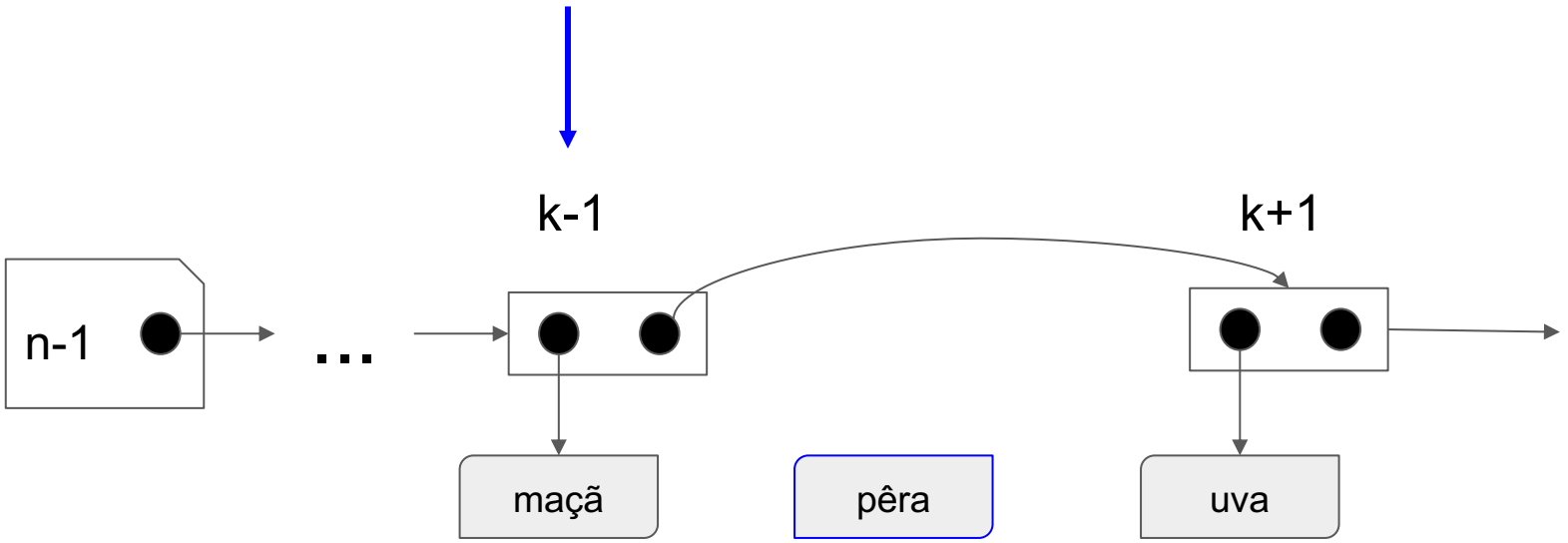
Remover o k-ésimo dado da lista, k>1

1. Acesse o elemento k-1
2. O elemento k-1 deve apontar para o próximo do elemento k (talvez nulo)
3. Desalocar o elemento k
4. Decrementar o tamanho da lista
5. Retornar o dado que estava no elemento k



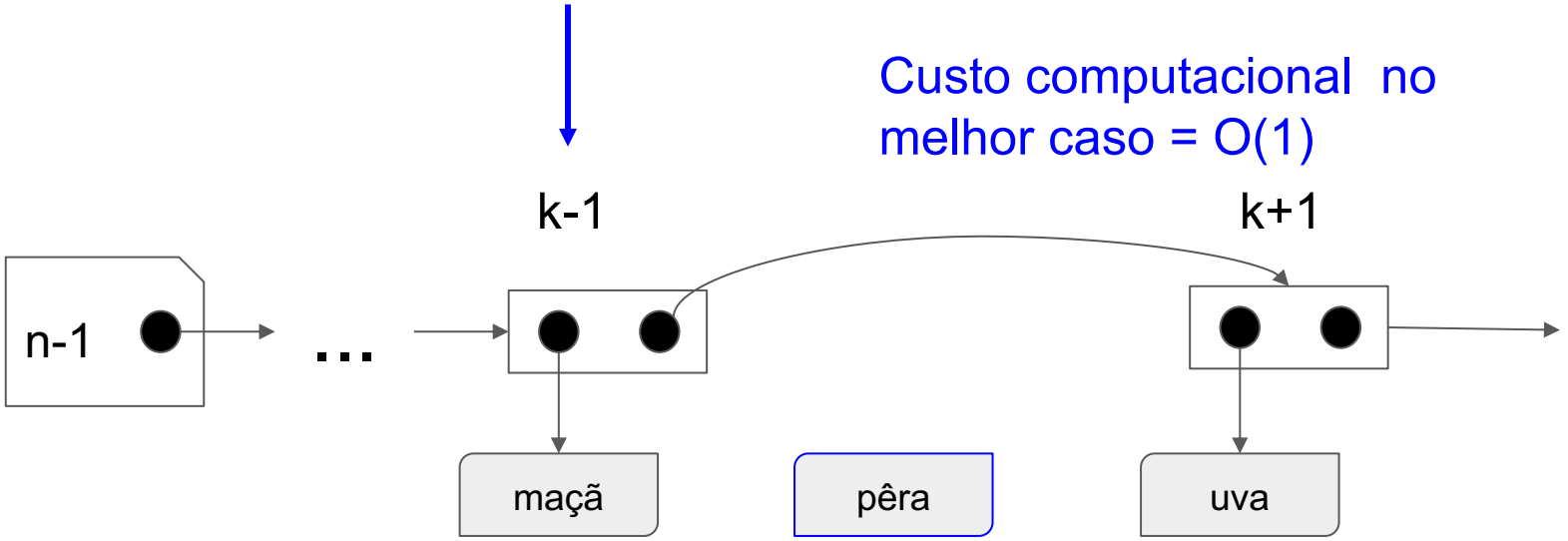
Remover o k-ésimo dado da lista, k>1

1. Acesse o elemento k-1
2. O elemento k-1 deve apontar para o próximo do elemento k (talvez nulo)
3. Desalocar o elemento k
4. Decrementar o tamanho da lista
5. Retornar o dado que estava no elemento k



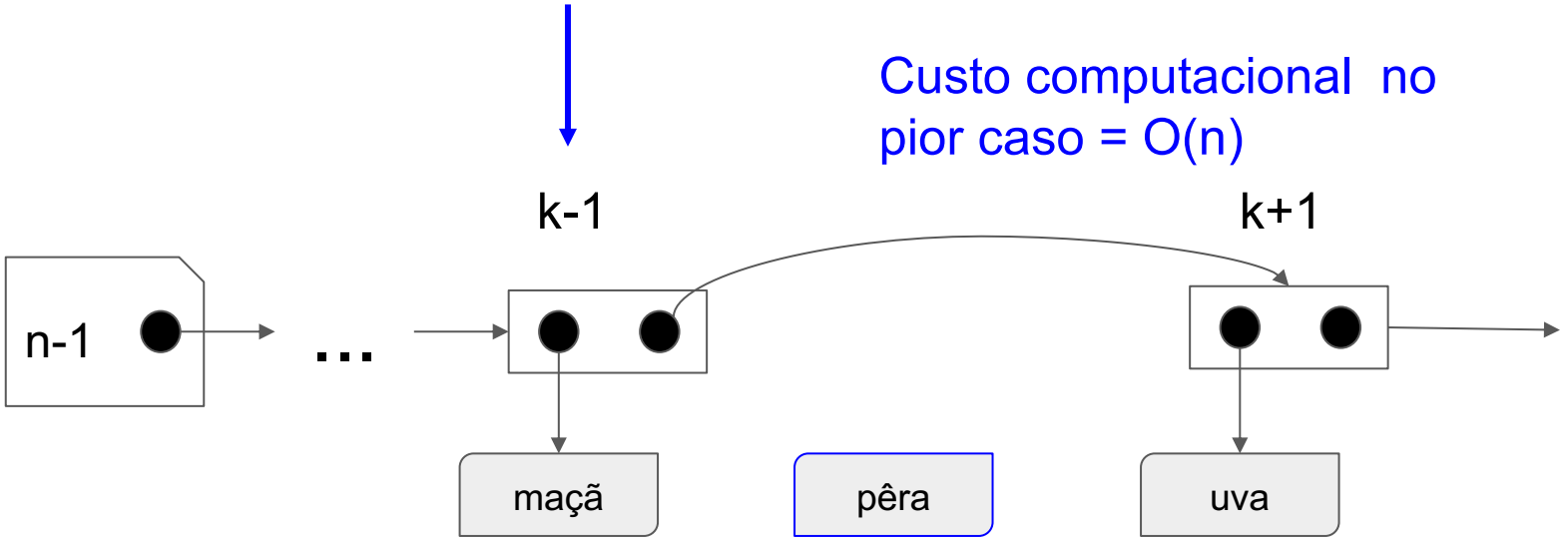
Remover o k-ésimo dado da lista, k>1

1. Acesse o elemento k-1
2. O elemento k-1 deve apontar para o próximo do elemento k (talvez nulo)
3. Desalocar o elemento k
4. Decrementar o tamanho da lista
5. Retornar o dado que estava no elemento k



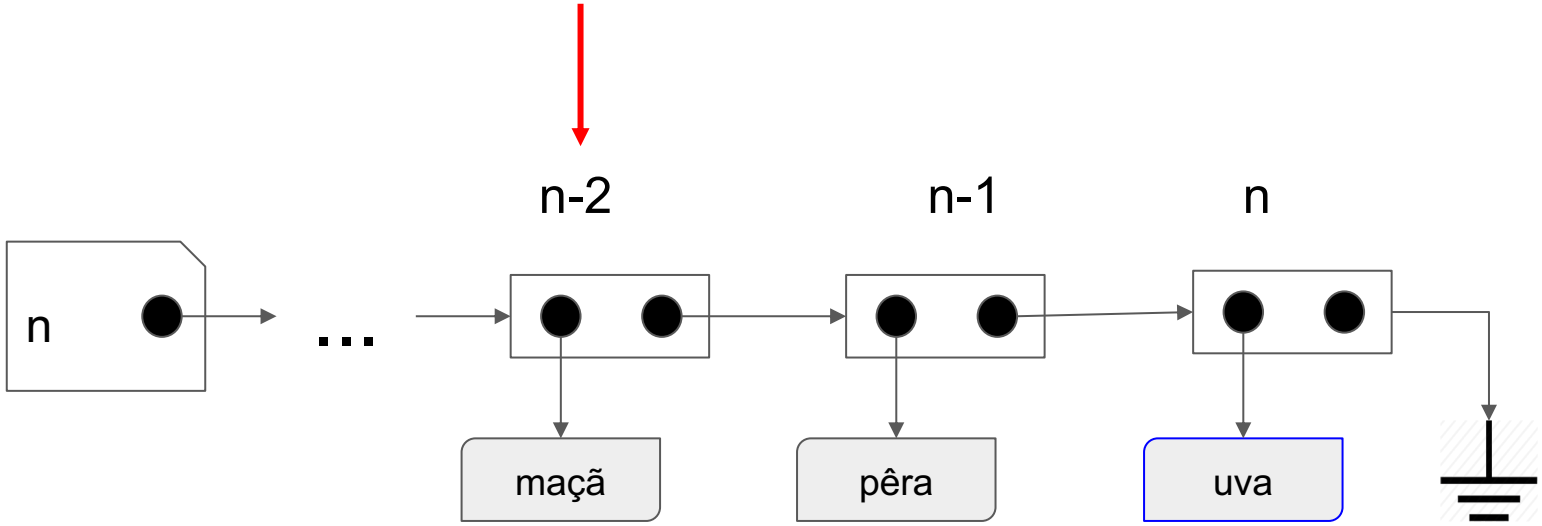
Remover o k-ésimo dado da lista, k>1

1. Acesse o elemento k-1
2. O elemento k-1 deve apontar para o próximo do elemento k (talvez nulo)
3. Desalocar o elemento k
4. Decrementar o tamanho da lista
5. Retornar o dado que estava no elemento k



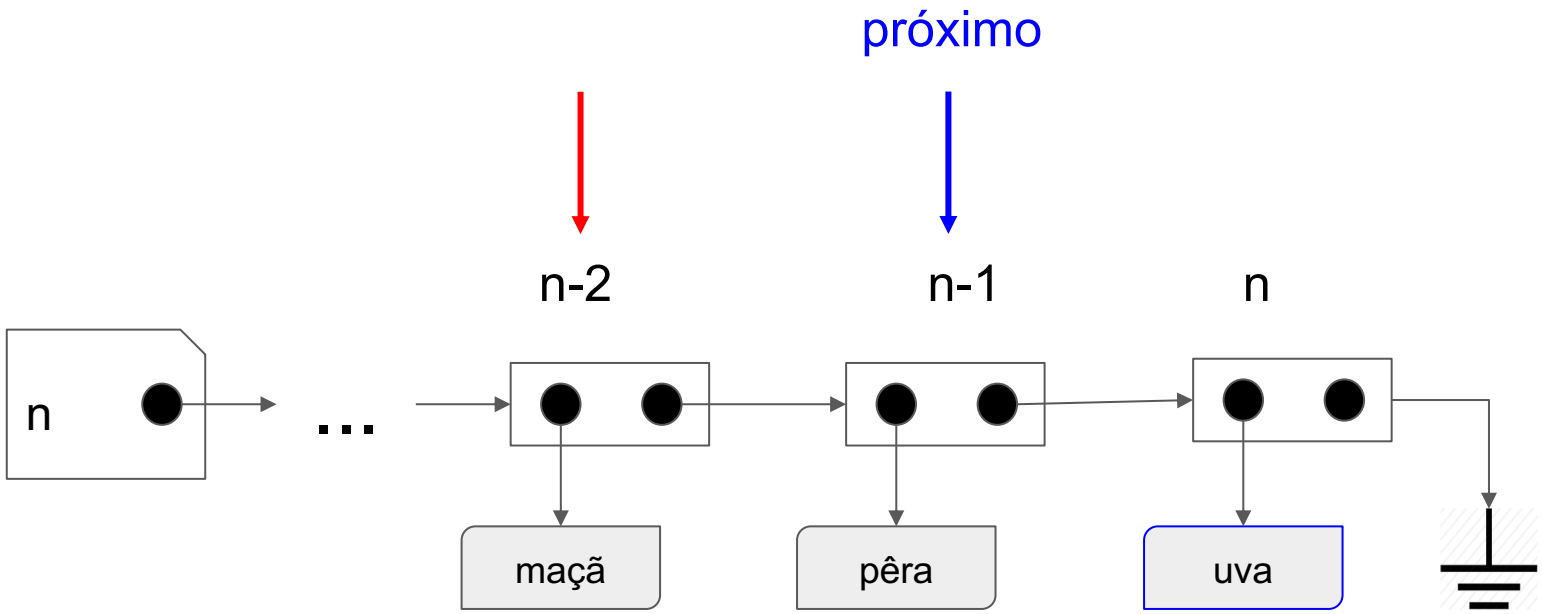
Destruir lista

1. Visitar elemento a elemento até fim da lista fazendo:
   a) Guardar referência para próximo elemento
   b) Desalocar o elemento atual
2. Desalocar registro de controle



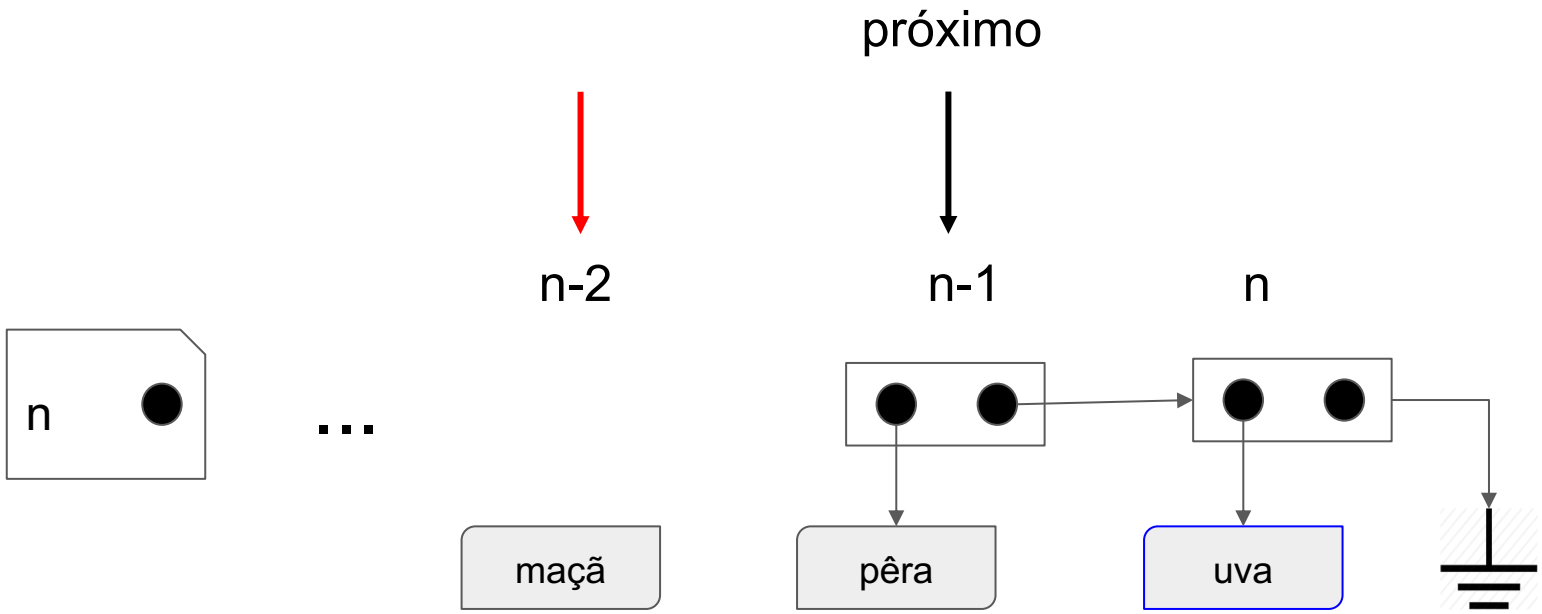
Destruir lista

1. Visitar elemento a elemento até fim da lista fazendo:
   a) Guardar referência para próximo elemento
   b) Desalocar o elemento atual
2. Desalocar registro de controle



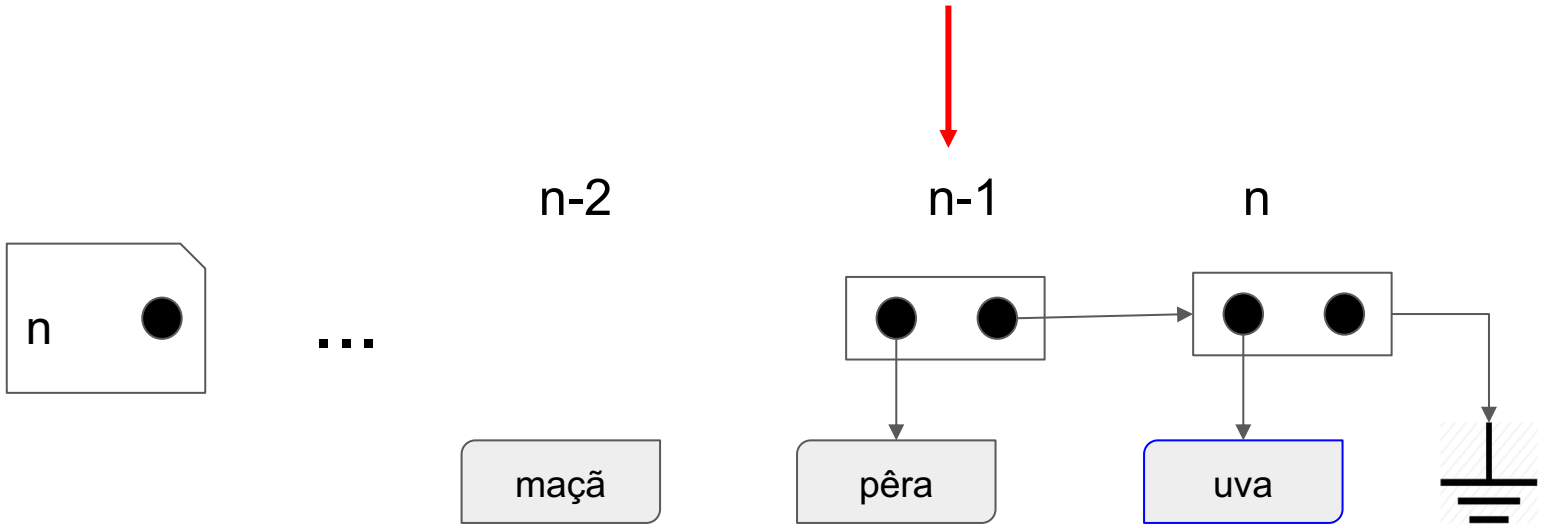
Destruir lista

1. Visitar elemento a elemento até fim da lista fazendo:
   a) Guardar referência para próximo elemento
   b) Desalocar o elemento atual
2. Desalocar registro de controle



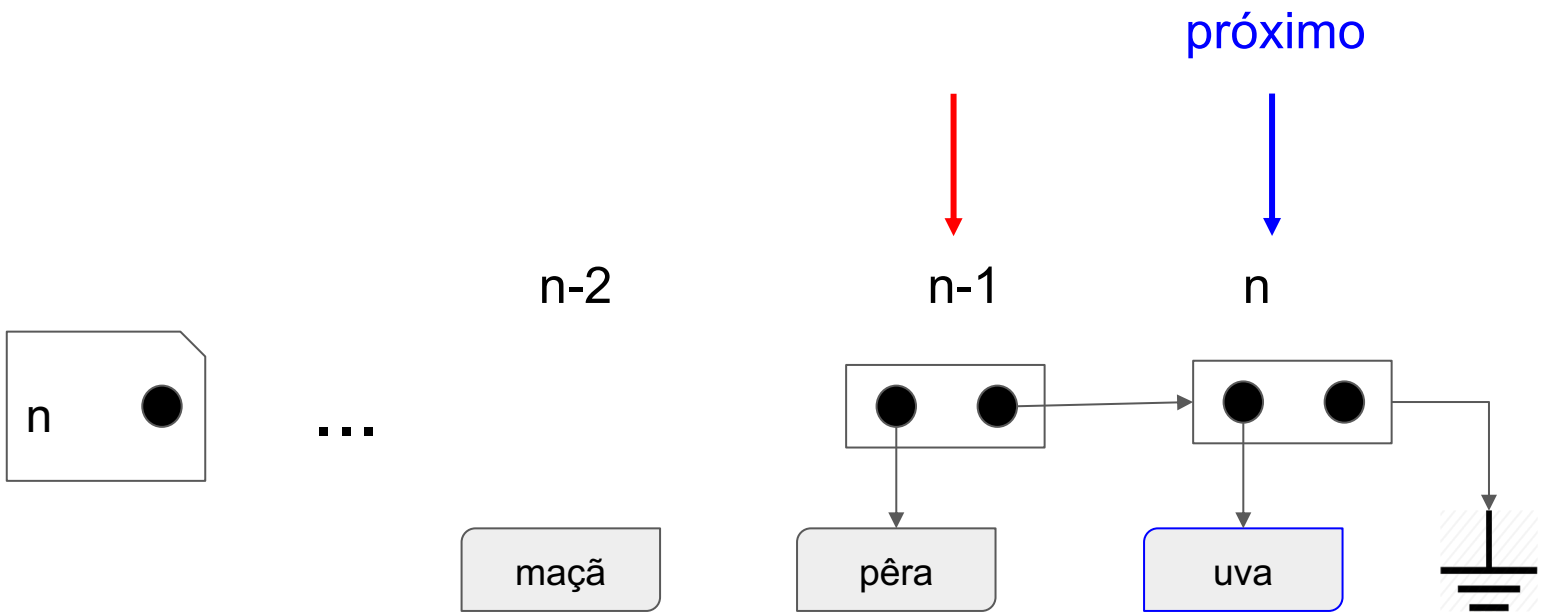
Destruir lista

1. Visitar elemento a elemento até fim da lista fazendo:
   a) Guardar referência para próximo elemento
   b) Desalocar o elemento atual
2. Desalocar registro de controle



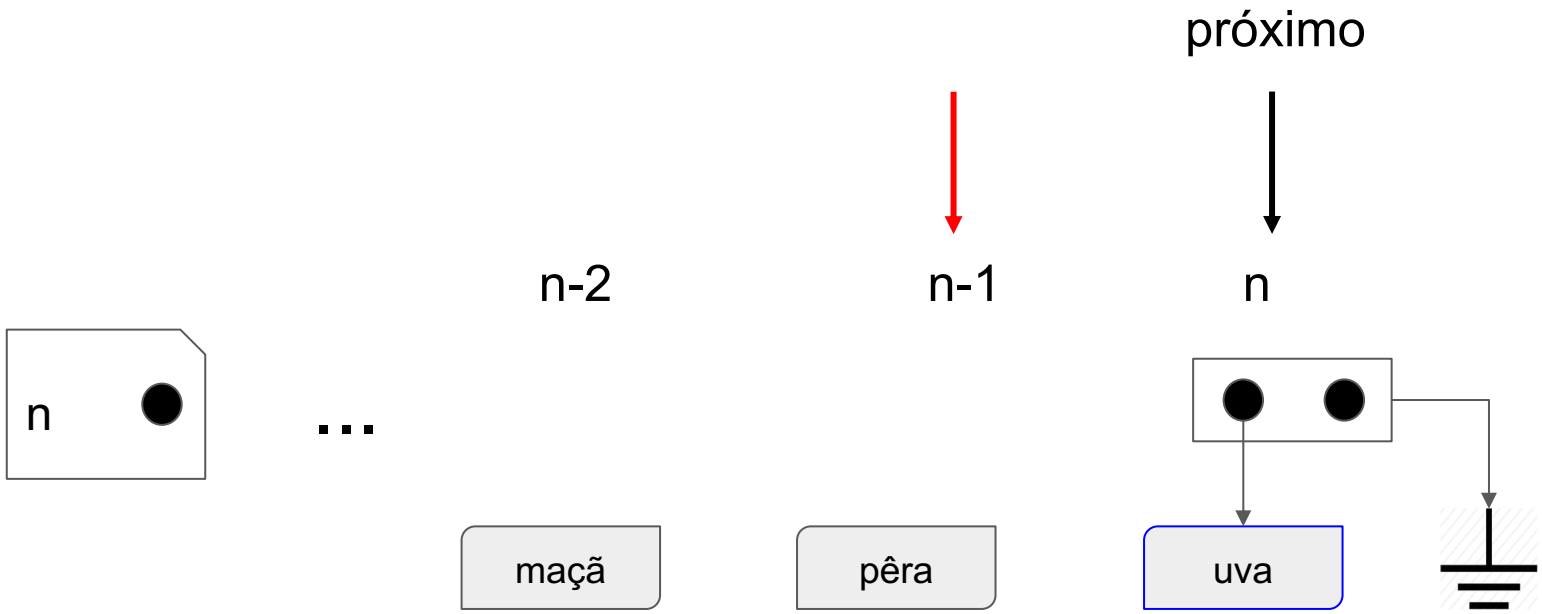
Destruir lista

1. Visitar elemento a elemento até fim da lista fazendo:
   a) Guardar referência para próximo elemento
   b) Desalocar o elemento atual
2. Desalocar registro de controle



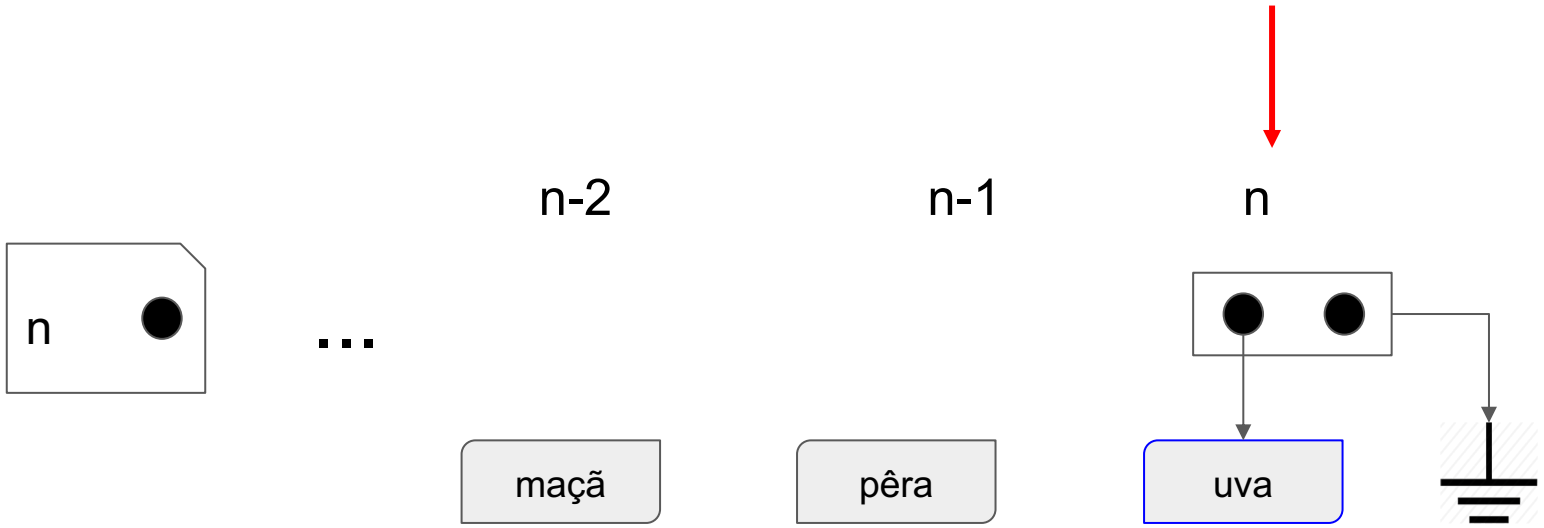
Destruir lista

1. Visitar elemento a elemento até fim da lista fazendo:
   a) Guardar referência para próximo elemento
   b) Desalocar o elemento atual
2. Desalocar registro de controle



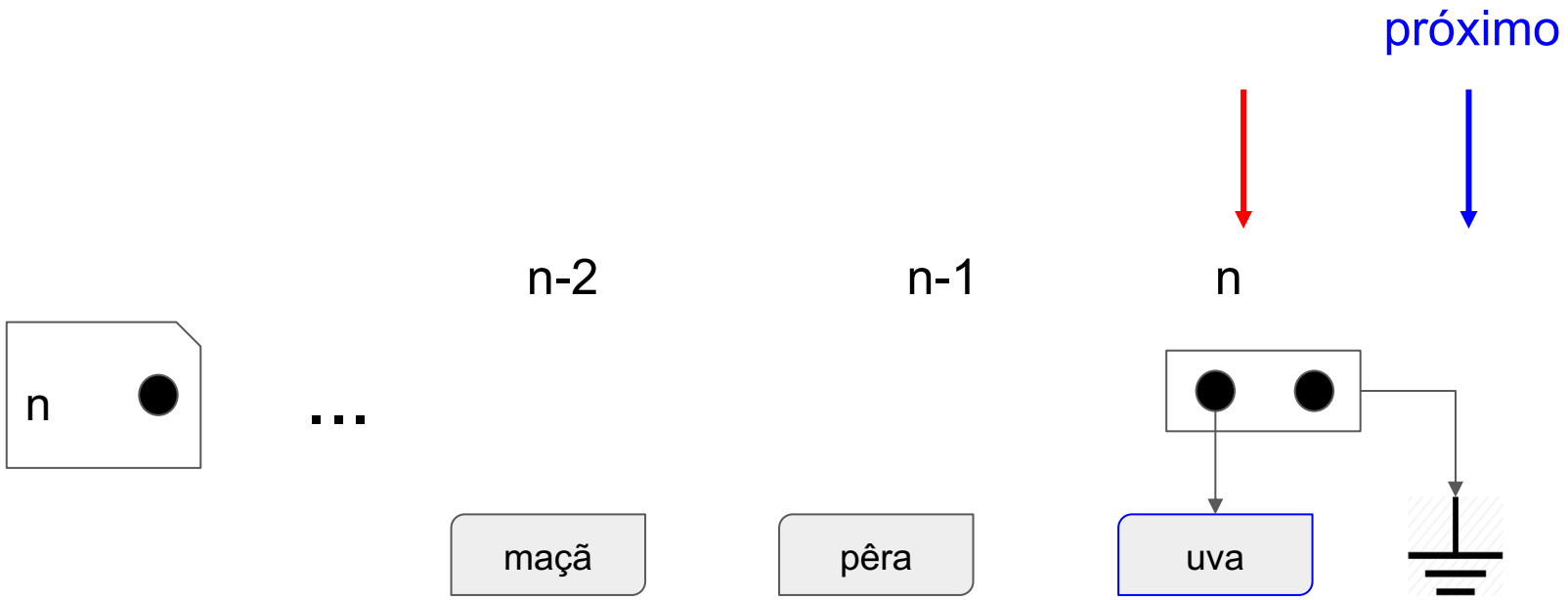
Destruir lista

1. Visitar elemento a elemento até fim da lista fazendo:
   a) Guardar referência para próximo elemento
   b) Desalocar o elemento atual
2. Desalocar registro de controle



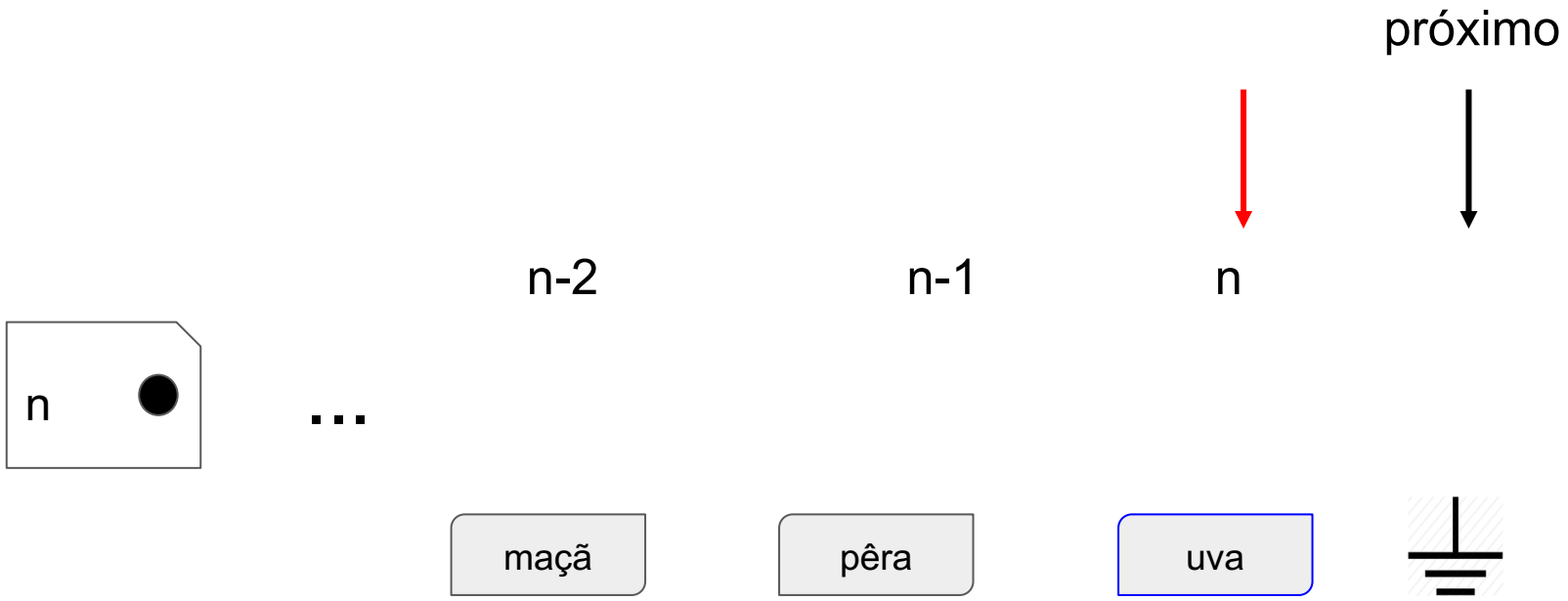
Destruir lista

1. Visitar elemento a elemento até fim da lista fazendo:
   a) Guardar referência para próximo elemento
   b) Desalocar o elemento atual
2. Desalocar registro de controle



Destruir lista

1. Visitar elemento a elemento até fim da lista fazendo:
   a) Guardar referência para próximo elemento
   b) Desalocar o elemento atual
2. Desalocar registro de controle

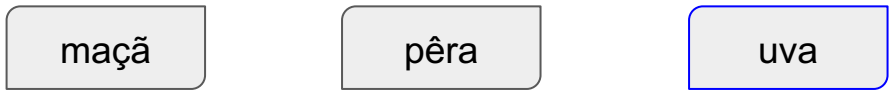


Destruir lista

1. Visitar elemento a elemento até fim da lista fazendo:
   a) Guardar referência para próximo elemento
   b) Desalocar o elemento atual
2. Desalocar registro de controle

Destruir lista

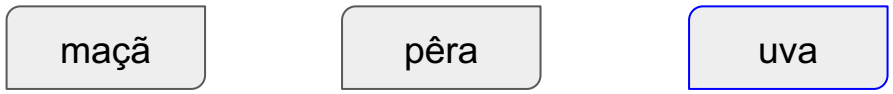
1. Visitar elemento a elemento até fim da lista fazendo:
   a) Guardar referência para próximo elemento
   b) Desalocar o elemento atual
2. Desalocar registro de controle



Destruir lista

1. Visitar elemento a elemento até fim da lista fazendo:
   a) Guardar referência para próximo elemento
   b) Desalocar o elemento atual
2. Desalocar registro de controle

Custo computacional = O(n)



Síntese da aula

- Como vetores, listas lineares são **sequências de dados**
- Diferente de vetores, listas lineares são **estrutura de dados dinâmicas**
- A **lista encadeada simples** é uma lista linear onde
  - cada **elemento** aponta para um **dado** e para o **próximo elemento** (talvez nulo)
  - a lista identifica a **quantidade de elementos** e aponta para o **primeiro elemento** (talvez nulo)
  - a partir do **primeiro elemento, visitam-se** os elementos **sequencialmente** em um **único sentido**
- A **complexidade** de acessar, inserir e remover nas listas encadeadas simples é **linear** O(n) no pior caso.