アルゴリズムとデータ構造a
6 - リスト構造

大見　嘉弘

今日の授業

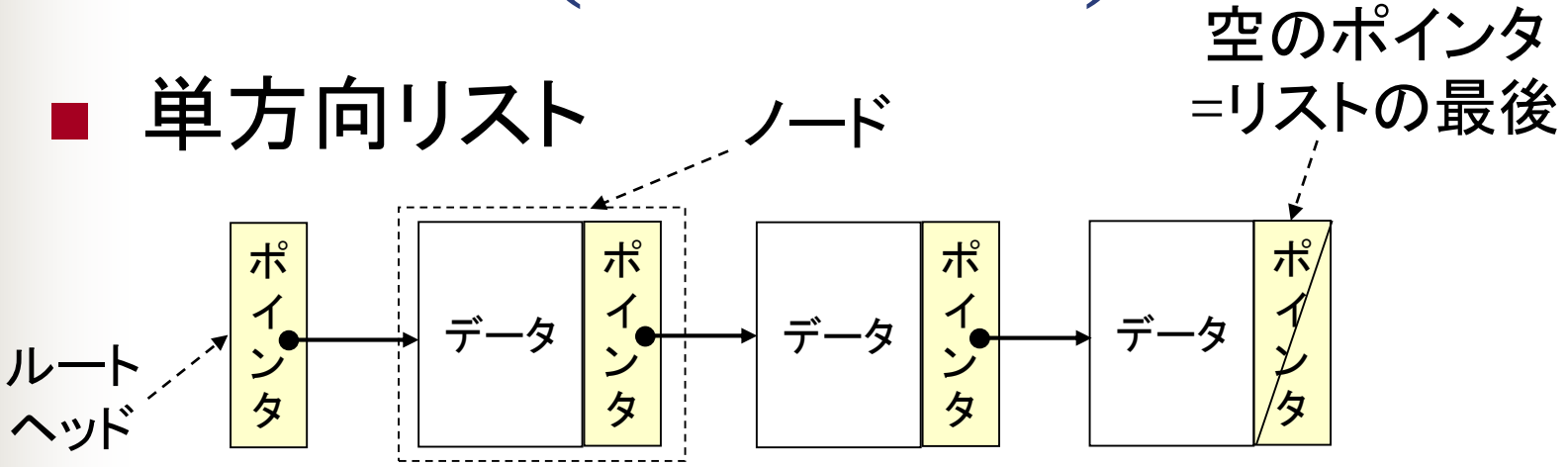
- リスト構造
  - 連結リスト
  - リストの操作

リスト構造

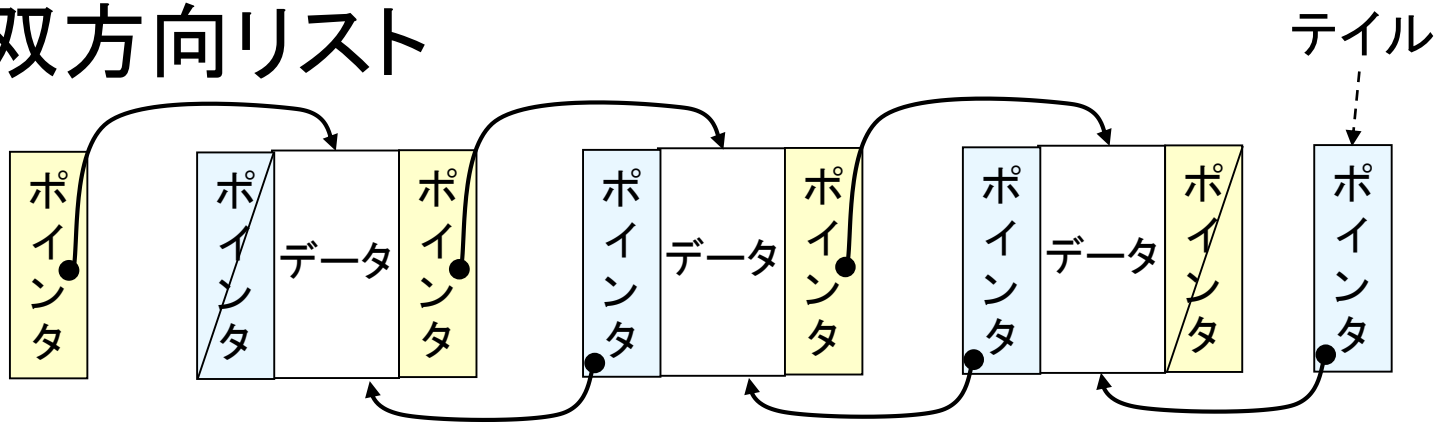
- リスト＝データの並び(一次元)
- データの並びを扱うデータ構造
  - 配列
  - リスト(連結リスト、線形リスト)
    ポインタで連結された構造
    通常、リストといえば連結リスト(線形リスト)
    を指す

連結リスト(線形リスト)の種類

- 単方向リスト



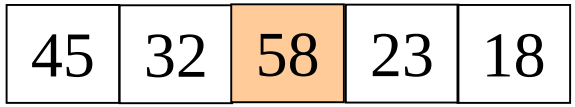
- 双方向リスト



配列の操作(探索、更新)

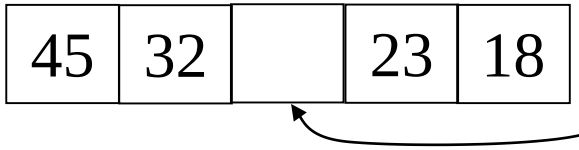
- 探索
  配列の添字を1,2,3…と1つずつ増やし、データが見つかったら完了。
  データの場所が事前に分かっている場合は、添字で直接指定する。
- 更新
  探索を行い、見つかったらその場所を新しいデータに書き換える。

配列における操作(削除)



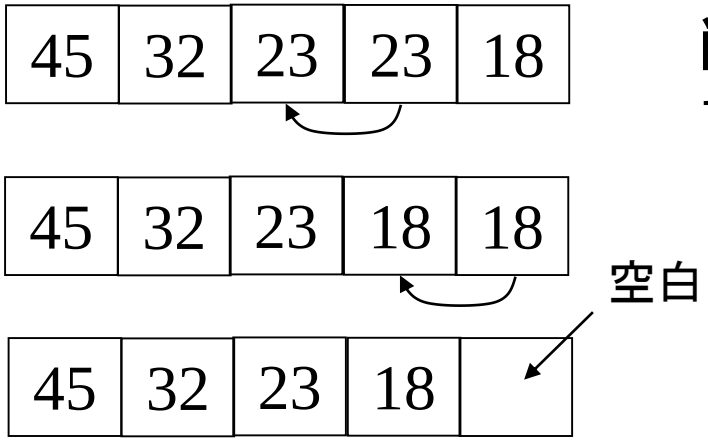
← 3番目のデータ(58)
を削除したい

- 論理的削除



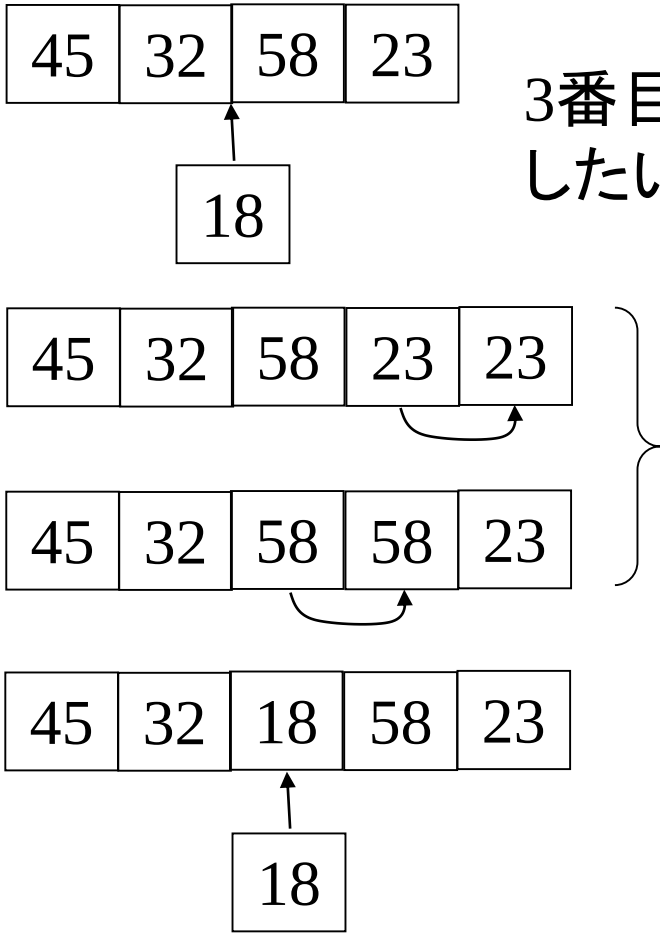
何もデータが入っていないという状態にする
(例:空白)

- 物理的削除



削除したデータの後ろを
1つずつ左に移動させる

配列における操作(挿入)



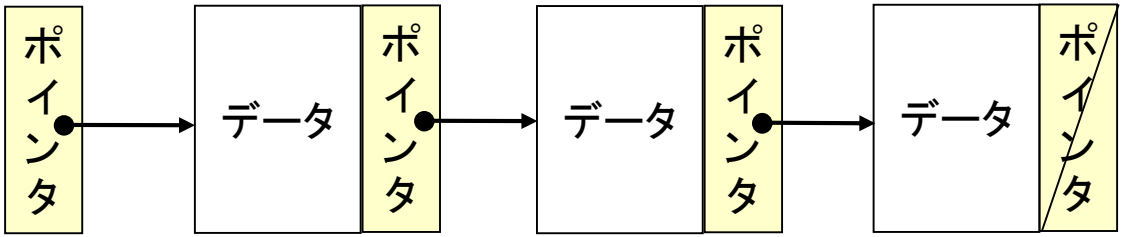
3番目に18というデータを挿入したい場合

挿入したい場所の後ろを1つずつ右に移動させる

データを挿入する

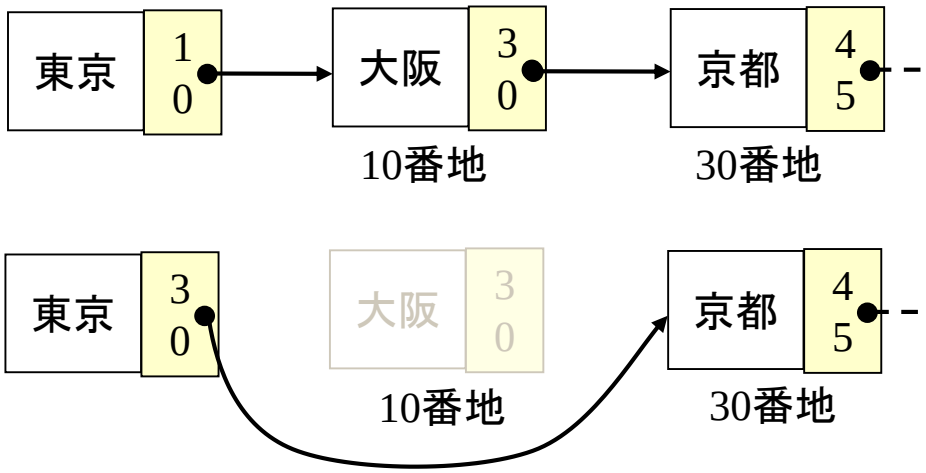
リストの操作(探索、更新)

- 探索
  先頭から順番にポインタをたどり探索。
  配列のような添字でのアクセスはできない。



- 更新
  探索を行って、そのデータの内容を新しいものに書き換える。リストの構造は変化しない。

リストの操作(削除)

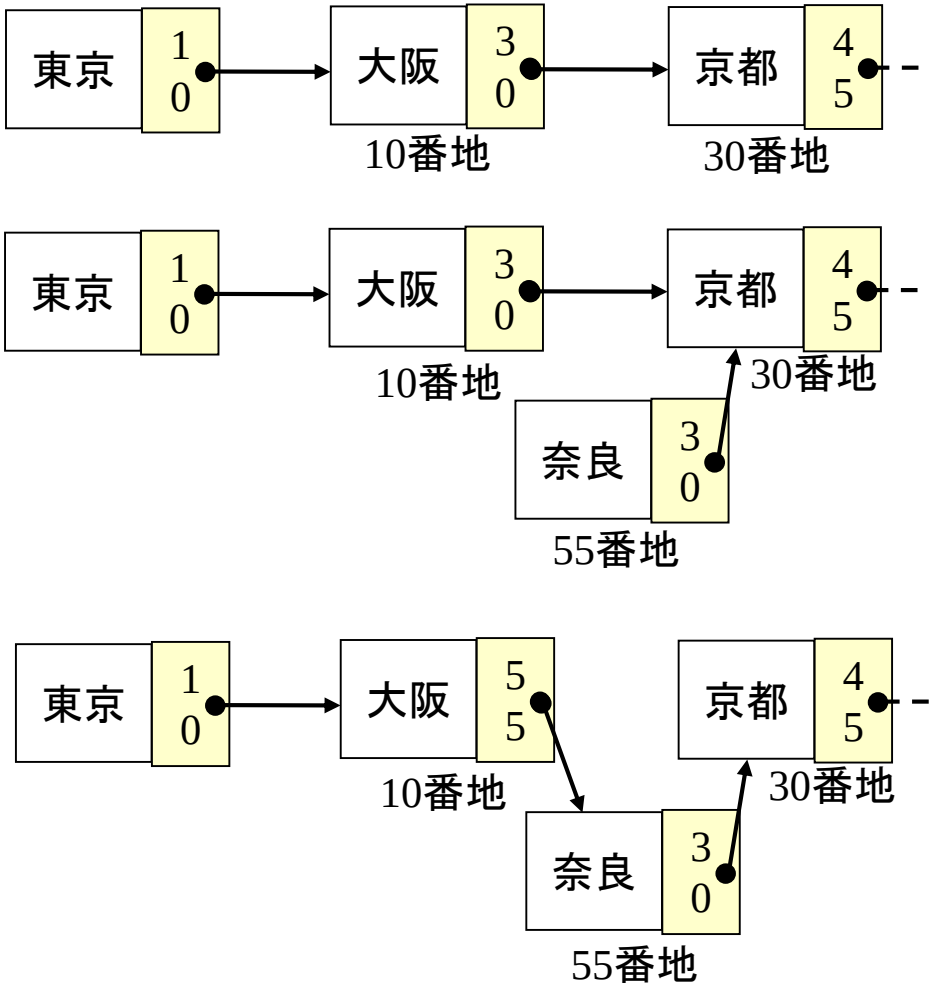


大阪を削除

大阪のノードは、メモリ上に残ってしまう(ガベージ)。

リストを物理的に再構成して、ゴミを消す(ガベージコレクション)。

リストの操作(挿入)



大阪と京都の間に奈良を挿入

リストと配列の比較

- 探索、更新
  データの場所が分かっている場合、配列が速い
- 削除
  配列の物理的削除は遅い
- 挿入
  配列の挿入は遅い

宿題

25	16	58	68	6

1. 上記の配列から68を物理的削除する様子を図示せよ。
2. 1.の結果の配列に10を挿入する様子を図示せよ。ただし、左から2番目(25と16の間)に挿入すること。