[B6] リスト処理

大崎 博之

関西学院大学 理工学部 情報科学科

ohsaki@kwansei.ac.jp

[B6] リスト処理

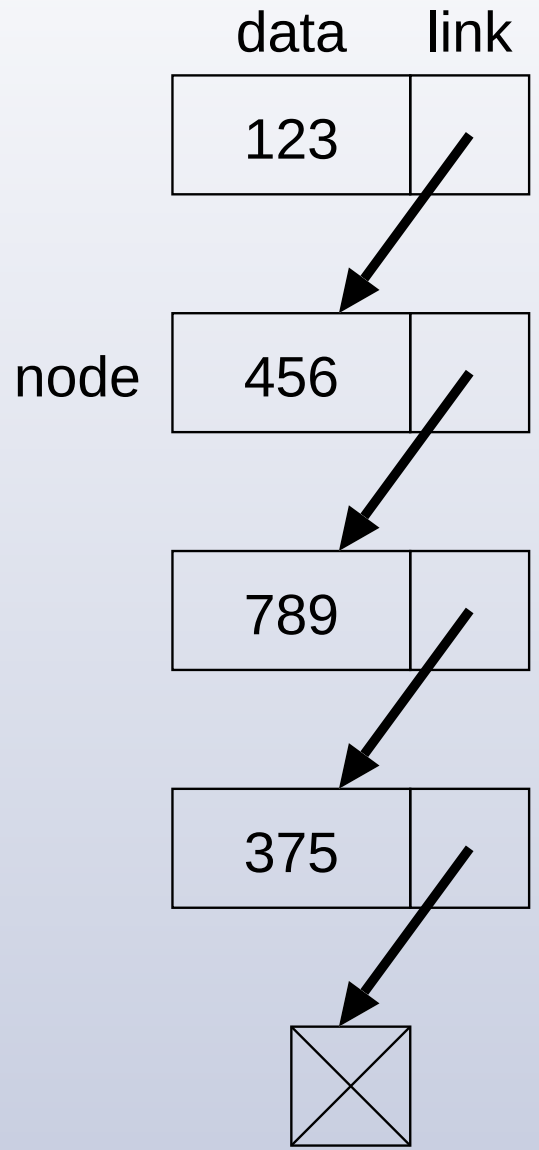
- 1. リスト
- 2. 二分木

これらを学ぶことに何の意味があるのか?

リスト (list)

コンピュータサイエンスにおける、もっとも基本的なデータ構造の一つ

- **ノード (node) の順列 (sequence)**
- **ノードとノードが連結 (link) されている**



リストの利点と欠点

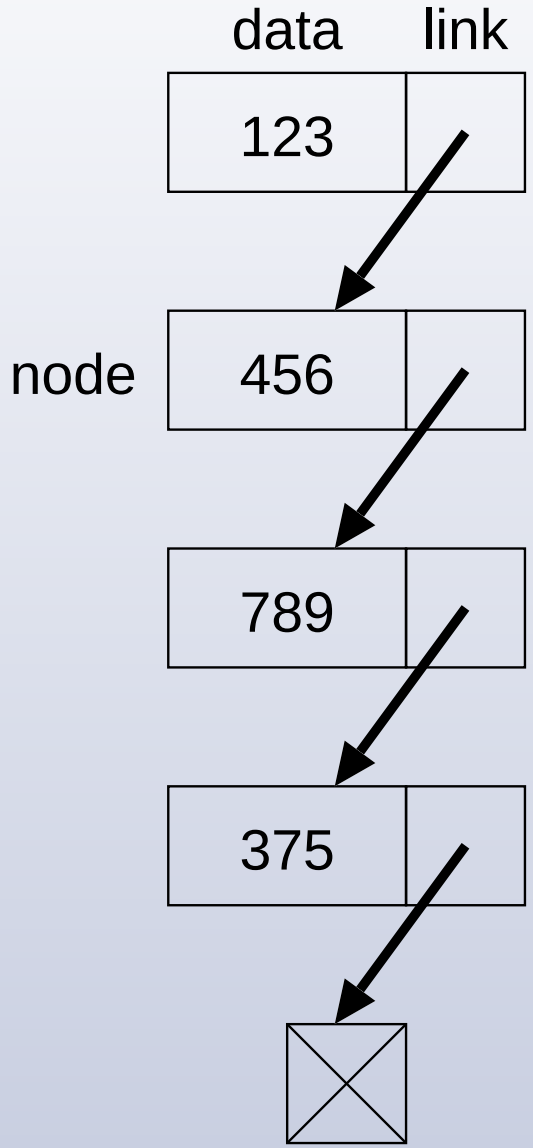
(配列と比較した時の) リストの利点

- **要素を自由に挿入・削除できる**

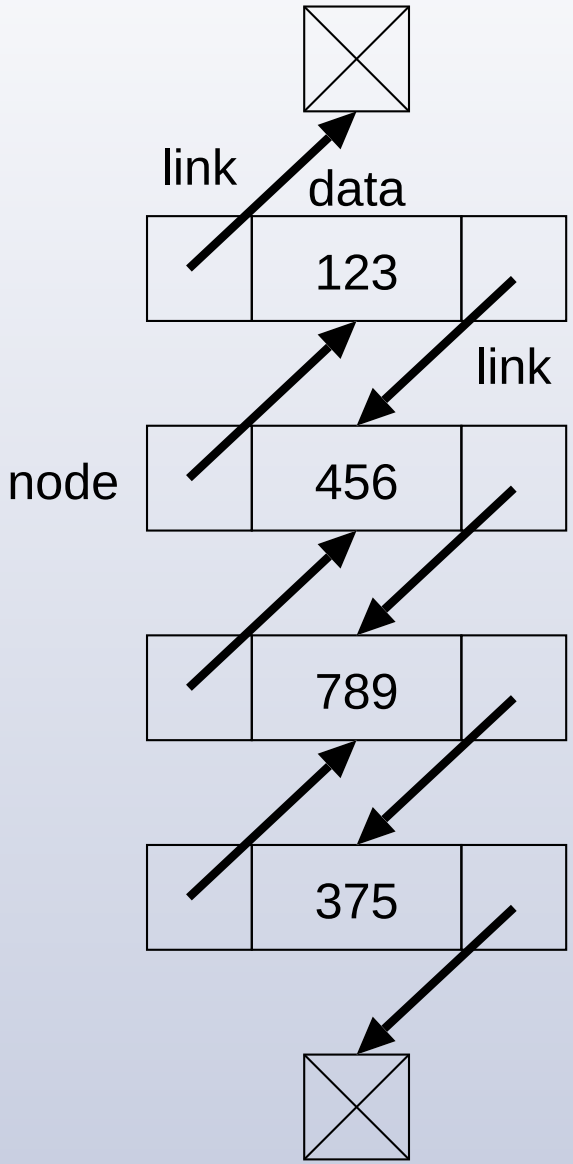
(配列と比較した時の) リストの欠点

- **各要素を参照するのが面倒 (リストを辿らないとダメ)**
- **途中のノードに直接アクセスできない**

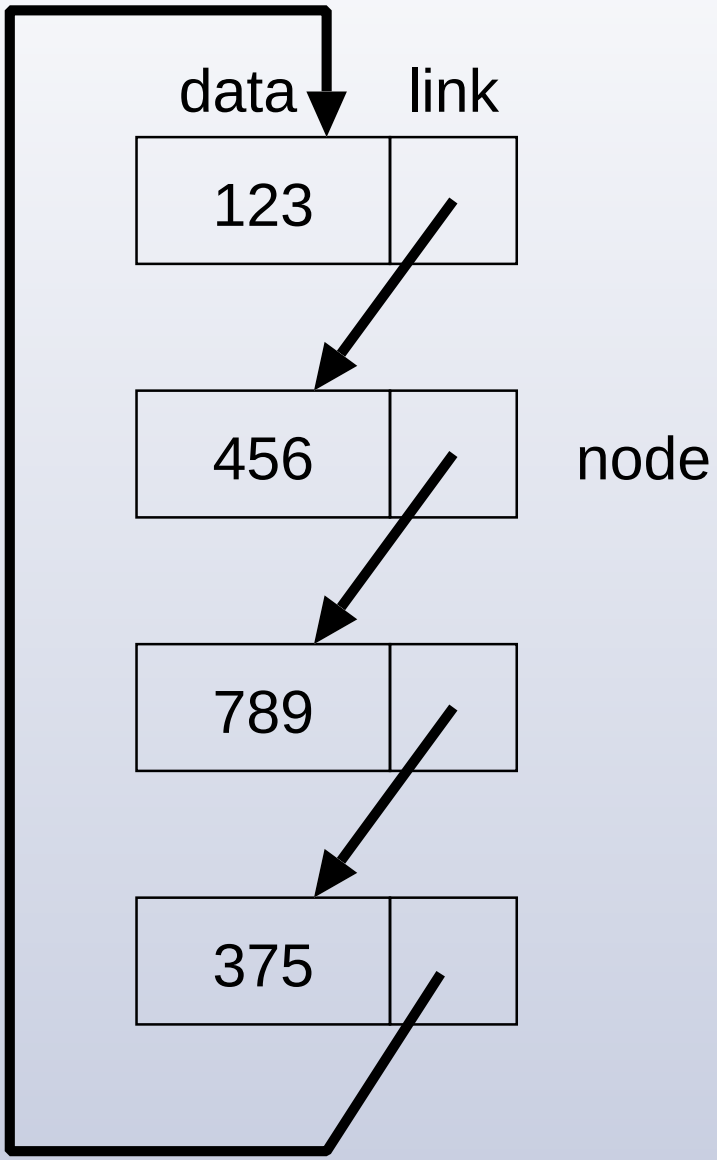
代表的なリスト: 片方向リスト (singly linked list)



代表的なリスト: 双方向リスト (doubly linked list)



代表的なリスト: 循環リスト (circular list)



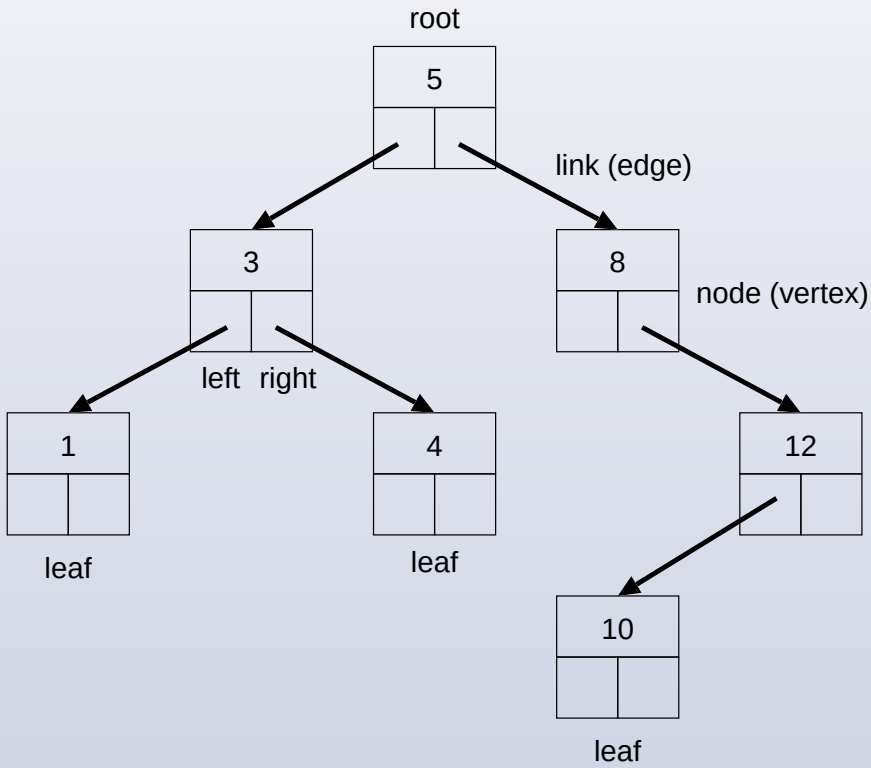
二分木 (binary tree)

コンピュータサイエンスにおける、基本的なデータ構造の一つ

- ノード (node) の木 (tree)
- ノードは 2 つの子ノードを持つ (左 (left)、右 (right))

高速なアルゴリズムが実現できるのはツリーのおかげ

- ノード数 N の二分木は深さ $\log_2 N$



リストやツリーはなぜ必要か?

- プログラミングの本質は アルゴリズム と データ構造
  - プログラミングの成功/失敗は……
    - 良い アルゴリズム を考えられるか?
    - 良い データ構造 を考えられるか?
  - プログラミングテクニックは実は瑣末 (さまつ) な問題
  - だから大学の講義はとても重要!

→ 設計した データ構造 を実現するために リスト や ツリー が不可欠