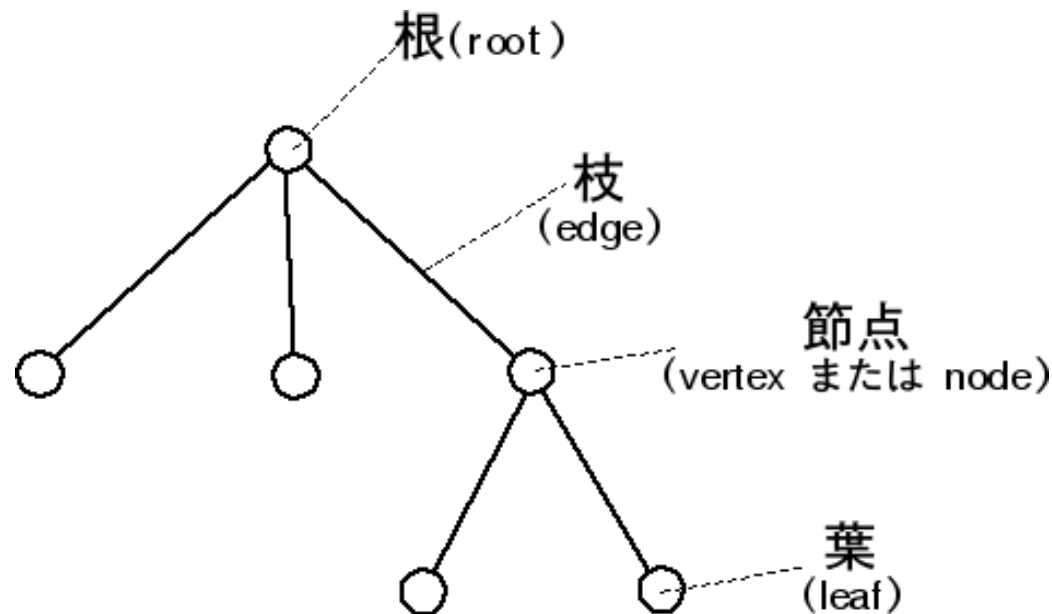


木とは

- 節点(node)と節点同士を結ぶ辺(edge)で表されるデータ構造
- 節点は円、辺は線で表される

根付き木

- 根(root)と呼ばれる他と区別された1つの節点を持つ木を根付き木(rooted tree)と呼ぶ
- 根付き木には親子・兄弟関係がある
- 子を持たない節点を葉(leaf)とよぶ



根付き木

- 根付き木の節点 x の子の数を x の次数(degree)とよぶ
- 根 r から節点 x までの経路の長さを x の深さ(depth)とよぶ
- 節点 x から葉までの経路の長さの最大値を節点 x の高さ(height)と呼ぶ

根付き木

- ALDS1_7_A: Rooted Trees

二分木

- 1つの根をもちすべての節点についてその子の数が2以下である木を根付き二分木(rooted binary tree)という
- 左の子、右の子を区別

二分木

- ALDS1_7_B: Binary Tree

二分木

- 二分木Tは再帰的に定義でき、以下のいずれかを満たす木
- Tは節点を全く持たない
- Tは共通要素を持たない次の3つの頂点集合から構成される:
 - 根(root)
 - 左部分木(left subtree)と呼ばれる二分木
 - 右部分木(right subtree)と呼ばれる二分木

木の巡回

- 根→左部分木→右部分木の順で節点を巡回する。
これを木の先行順巡回(Preorder Tree Walk)とよぶ
- 左部分木→根→右部分木の順で節点を巡回する。
これを木の間順巡回(Inorder Tree Walk)とよぶ
- 左部分木→右部分木→根の順で節点を巡回する。
これを木の後行順巡回(Postorder Tree Walk)とよぶ

木の巡回

- ALDS1_7_C: Tree Walk

木の復元

- 先行順巡回と中間順巡回から、二分木を復元して、後行順巡回を出力する問題を考える
- 先行順巡回は根→左部分木→右部分木、中間順巡回は左部分木→根→右部分木の順に巡回する
- 例えば先行順巡回が{1,2,3,4,5,6}、中間順巡回が{3,2,1,5,4,6}のとき、全体の根は1、左部分木は{3,2}、右部分木は{5,4,6}であることがわかる

木の復元

- ALDS1_7_D: Reconstruction of the Tree

おしまい

- おつかれさまでした