

配列

- 直接データにアクセスすることができる
- 連続したデータの途中などにデータを追加するとき、一番最後尾の添字の中身を一つ後にずらし、同じようにデータを追加しようとする添字まで同じように他の要素もずらして、最後に指定した添字に該当するデータを追加する
 - そのため配列での削除や追加は非効率である
- 連結リスト
-

単方向リスト

- リストはノードの集まりでできている
 - ノードの構成：データ・ポインタ
 - データ：格納している情報のこと
 - ポインタ：移動するノードを指し示す
- オブジェクトの中には以下のような形で格納されている [[ノード・データ・ポインタ],[ノード・データ・ポインタ],[ノード・データ・ポインタ]]
- 特徴1：先頭から特定のノードを順番に探す
 - デメリット：構成するリストの途中などにアクセスしようとしても、必ず頭から順にアクセスしていく
 - 後ろからのアクセスはできない
- 特徴2：リストの追加が簡単
 - 変更に関連するノード間のポインタの変更でノードの追加と削除ができ、容易

双方向リスト

- リストの一種
- ノードの構成：前ポインタ・データ・後ポインタ
- 調べてはないがおそらくオブジェクトにノードを格納していくときの形はほとんど単方向と一緒
- 特徴：先頭と末尾どちらからでもアクセスが可能になっている

※ 配列・リストどちらも内部的な動作をそれぞれで持ち合わせている

ツリー構造

2分探索木：探索しやすい木構造

ルール：

左の子 > 親 > 右の子