

宿題 1.コンテナの種類と特徴

基本的な情報

- `bitset`
固定長配列, bit 値を格納
- `array`
固定長配列, 様々な型の値を格納
- `vector`
可変長配列, 高速, メモリ効率が良い
- `deque`
可変長配列, 末尾・先頭への挿入・削除が高速
- `list`
前後の要素同士を結合した双方向連結リスト
- `forward_list`
前方のみと結合した片方向連結リスト
- `set`
要素の大小関係を自動で並び替え, 分木に格納。大小のある型に対応, 要素自体がキー
- `multiset`
多重集合 (set と異なり, 重複するデータの保持が可能)
- `unordered_set, unordered_multiset`
要素をキーのハッシュ値に基づくハッシュテーブルに格納, 順序なし
(`unordered_set`: 重複なし, `unordered_multiset`: 重複あり)
- `map`
連想配列, 要素を検索可能なキー(様々な型が可能)と紐付け
- `multimap`
連想配列, ひとつのキーに複数の対応する要素をもつことが可能
- `unordered_map, unordered_multimap`
連想配列, キーと要素のペアをハッシュで管理, 順序なし

ある要素への番号（またはキー）を指定したアクセスにかかる計算量

型	計算コスト (n は格納された要素数)
vector	$O(1)$
deque	$O(1)$
unordered_map, unordered_multimap	$O(1) \sim O(n)$
map, multimap	$O(\log n)$
list, forward_list	$O(n)$

要素の挿入・削除にかかる計算量

型	計算コスト (n は格納された要素数)
vector	$O(1) \sim O(n)$ (* 1)
deque	$O(1) \sim O(n)$ (* 2)
list, forward_list	$O(1)$
set, multiset	$O(\log n)$
unordered_set, unordered_multiset	$O(1) \sim O(n)$

(* 1) :

末尾への削除, また末尾への挿入 (平均的な場合) $\cdots O(1)$

末尾への挿入 (もっともコストの高い場合) $\cdots O(n)$

上記以外への挿入・削除 $\cdots O(n)$

(* 2) :

先頭・末尾への削除, また先頭・末尾への挿入 (平均的な場合) $\cdots O(1)$

先頭・末尾への挿入 (もっともコストの高い場合) $\cdots O(n)$

上記以外への挿入・削除 $\cdots O(n)$

要素の検索にかかる計算量

型	計算コスト (n は格納された要素数)
unordered_set, unordered_multiset	$O(1) \sim O(n)$
set, multiset	$O(\log n)$
vector	$O(n)$
deque	$O(n)$
list, forward_list	$O(n)$
map, multimap	$O(n)$
unordered_map, unordered_multimap	$O(n)$