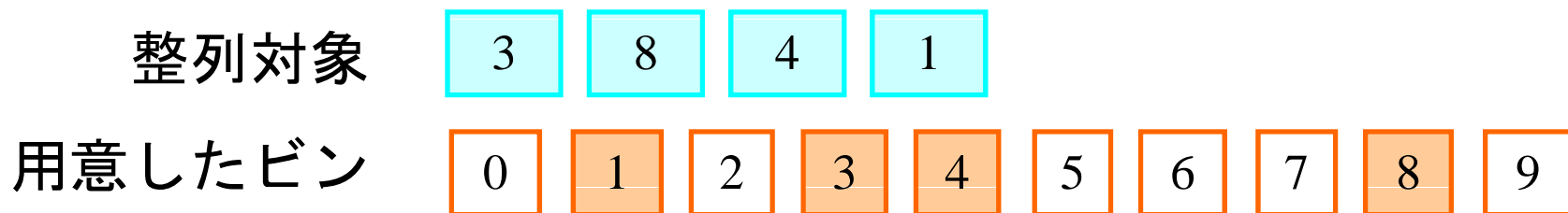


ビン（バケツ，バケツ）ソート

- (1) 各キー値に対して，空の入れ物（ビン）を用意する。
- (2) 整列対象の要素値に対して，キー値に対応するビンに要素値を入れ，「空でない」の印を付ける。
- (3) 入れ物を走査して，「空でない」が付いた要素を集める。

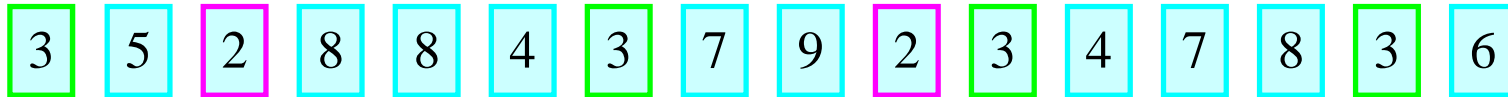
問題点： 予め，キー値の範囲が分かっている必要がある。
キー値に重複があると，要素が失われる。
→ ビンに，要素を蓄えるリストを用意。



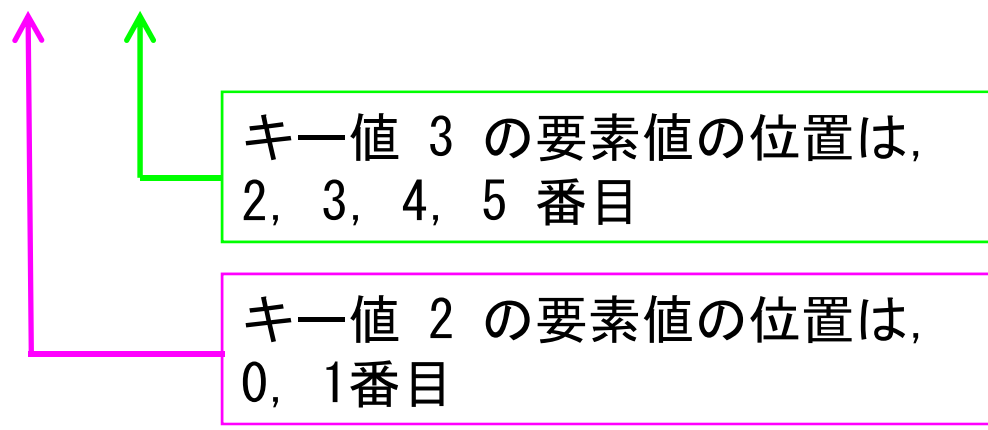
分布数え上げソート

- (1) 各キー値に対して, カウンタを用意する.
- (2) 全てのカウンタを 0 にする.
- (3) 整列対象の各要素に対して,
 キー値に対応するカウンタを, 1 増やす.
 (キー値 key_i の度数分布 $freq_i$ ができる)
- (4) 全てのキー値 key_i に対して,
 キー値の累積度数 cum_i を求める.
- (5) 整列対象の要素に対して,
 キー値の度数分布, 累積度数に従い, 要素の位置を
 決定し, 要素を配置する.

分布数え上げソートの実行



キー値	0以下	1	2	3	4	5	6	7	8	9	10以上
度数分布	0	0	2	4	2	1	1	2	3	1	0
累積度数	0	0	2	6	8	9	10	12	15	16	16



ビンソート, 分布数え上げソートの計算量

要素をビンに入れる.

キー値ごとの要素数を数える.

→ $O(n)$

ビンに入れた要素を集める.

累積度数に従い要素番号を決定する.

→ $O(n)$

あわせて, $O(n)$

ビンソートは, 安定.

分布数え上げソートは, 工夫をすれば, 安定.

基数ソート

k 桁のキー値の整列において,

1 桁目を, 安定な方法で整列する.

2 桁目を, 安定な方法で整列する.

: : : :

k 桁目を, 安定な方法で整列する.

要素数 n の各桁の数値を整列する計算量を $S(n)$ とすれば,
全計算量は, $k \times S(n)$.

工夫したマージソートやヒープソートは,

$O(n \log_2 n)$ の安定な整列.

ビンソート, 工夫した分布数え上げソートは, 高速
かつ安定した整列.

基数ソートの実行

1桁目を整列

4 5 1	4 5 1	8 0 9	0 3 2
0 3 2	0 3 2	0 3 2	3 5 2
5 4 8	7 3 2	7 3 2	4 5 1
8 0 9	3 5 2	5 4 8	5 4 8
5 7 3	5 7 3	4 5 1	5 7 3
7 3 2	5 4 8	3 5 2	7 3 2
3 5 2	8 0 9	5 7 3	8 0 9

安定な整列のため、1桁目で決定された前後関係は変わらない。

安定な整列のため、1, 2桁目で決定された前後関係は変わらない。