アルゴリズム B 演習 No. 4　　2015 年 10 月 30 日配布

4. 優先順位付きキュー，ヒープ，ハッシュ表

- **優先順位付きキュー**は，次の操作を持つデータ構造である．
  - insert(x): x を挿入する．
  - deletemin: 優先順位付きキューの最小の要素を返し，キューから削除する．

  優先順位付きキューは，ヒープを用いて実現できる．
- **（2 分）ヒープ**とは，次の条件を満たす 2 分木である．
  - 各頂点の値はその子の値以下である．
  - 深さ最大の段は左詰めになっており，それ以外の段はすべて埋まっている．

  ヒープの条件から，ヒープの根の値は最小値である．ヒープの上から順に左から右へ向かって添字を付けることで，ヒープを配列で表現できる[*1]．

  要素 x を挿入するには，まず配列としての末尾に x を追加する．x を 2 分木の頂点と見なしたときの親と値を比較し，x の方が値が小さいとき交換する．交換した場合は x の親と値を比較し同様に繰り返せば最終的にヒープの条件を満たす．

  最小値を削除するには，まずヒープの根を削除し，配列と見なして末尾にあたる頂点 x を根に移動する．x が子を持つとき，x の子で x より値が小さいものがあれば，そのうちの小さい方と x を交換する．この操作を繰り返して交換が起きなくなったとき，ヒープの条件を満たしている．
- **ハッシュ表**は，順序構造を用いずに，要素の挿入，削除，探索がほぼ定数時間で行えるデータ構造である．要素の格納場所を決めるために，ハッシュ関数と呼ばれる関数を用いる．異なる要素に対するハッシュ値が等しくなる可能性があり，これを**衝突**という．衝突に対処する方法として以下のものがある．
  - **チェイン法** (chaining) では，ハッシュ値が等しい要素を連結リストに格納する．ハッシュ表には連結リストへのポインタが格納されている．
  - **開番地法** (open addressing) では，ハッシュ表そのものに要素を格納する．

    要素の挿入の際，ハッシュ値を計算し，ハッシュ表の対応する場所が空または削除済みのときその場所に格納する．要素が入っていた場合は次の場所を調べ，同様に繰り返す．

    要素の削除の際は，ハッシュ表の対応する場所に「削除済み」の印を付ける．

    要素の探索の際には，「削除済み」の場所は飛ばして次を調べる．

[*1] この配列をヒープと呼ぶこともある．

問題

4-1. 以下の配列で表現されたヒープを 2 分木として図示せよ．

添字	0	1	2	3	4	5	6	7
	35	36	72	50	38	99	73	95

4-2. ヒープで実装された空の優先順位付きキューに対し，次の操作を行ったときの過程と最終的に得られる木を図示せよ．

$$\text{insert}(68) \to \text{insert}(38) \to \text{insert}(44) \to \text{insert}(20) \to \text{insert}(30) \to \text{deletemin} \to \text{insert}(15)$$

4-3, 4-4 では，ハッシュ関数は 8 で割った余りを返すものとする．また，ハッシュ表の大きさは 8 であり，0 から 7 までの整数で位置を指定するものとする．

4-3. チェイン法を用いた空のハッシュ表に対し，次のように insert（要素の挿入）と delete（要素の削除）を行ったとき，最終的なハッシュ表を図示せよ．

$$\text{insert}(48) \to \text{insert}(84) \to \text{insert}(77) \to \text{insert}(60) \to \text{delete}(84) \to \text{insert}(69)$$

4-4. 開番地法を用いた空のハッシュ表に対し，次のように insert（要素の挿入）と delete（要素の削除）を行ったとき，最終的なハッシュ表を図示せよ．

$$\text{insert}(48) \to \text{insert}(84) \to \text{insert}(77) \to \text{insert}(60) \to \text{delete}(84) \to \text{insert}(69)$$

4-5. ヒープで実装された n 個の要素を持つ優先順位付きキューにおいて，insert を行ったときの頂点の交換回数が $O(\log n)$ であることを示せ．

4-6. チェイン法を用いた大きさ m のハッシュ表に n 個の要素を挿入する．どの要素についても独立に各ハッシュ値が等確率で現れるとする．このとき，各ハッシュ値に対応する連結リストの長さの期待値を求めよ．