Quiz02 - Binary Heap

1． 配列 A[1:n] が **maximum heap** になっているとする．
次のうち、配列 A 内で必ず成立する順序関係を全て選択せよ．

1. A[7] ≧ A[14] ○
2. A[8] ≧ A[9]
3. A[2] ≧ A[5] ○
4. A[3] ≧ A[13] ○
5. A[4] ≧ A[7]
6. A[1] ≧ A[5] ○

二分ヒープの定義：
親ノードと子ノードの間，つまり A[n] と A[$n/2$],
あるいは A[$2n$],A[$2n$+1] との間で与えられた大小関係が必ず成立する．
つまり，maximum heap であれば，任意の n において，A[$n/2$] ≧ A[n] の関係が成立していることを
意識すればよい．兄弟ノード A[$2n$],A[$2n+1$] 同士にはこの順序関係は無いので気を付けること．

2． 配列にヒープ構造を持たせることで，我々はどのような恩恵を受けるだろうか？ 下文の空欄を埋めよ．

わざわざ配列全体を[ソート]しなくても，ヒープ条件として定義された順序関係に基づいた
その配列中の要素のうちの[最大]値または[最小]値のみを容易に取り出すことができる．

最大値，最小値を配列から取り出すだけであれば，配列全体をソートをする必要は無い．
ヒープを利用すれば，$O(\log n)$ 程度の計算量で配列内から最大値，最小値を取得できる．
⇒ これを利用，繰り返しすことでソートを行う「ヒープソート」は，$O(n \log n)$ で実現できる．

3． 配列 A[1:7] = { 10, 7, 6, 5, 1, 4, 2 } は **maximum heap** になっている．ここに，講義資料中の insert() 関数を用いて 新しい要素 **8** を A に挿入しようとするとき，**どのような挙動を経てどんなアップデート後の maximum heap を得るか？** 講義資料に記載されたプログラムの手続きに忠実に従って説明せよ．

A[8] に新しい要素「8」を挿入すると，A[1:7] はヒープになっているが，A[1:8] はヒープとは限らない．
A[8]=8 に対して upheap() を適用すれば，A[4]=5, A[2]=7, A[1]=10 だが，
v=A[8]>A[4], v=A[8]>A[2] であるから，A[8]=A[4], A[4]=A[2] と値を移動させる．
v=A[8]<A[1] であるから，結局 A[2]=v となって，
新しい maximum heap は A[1:8] = { 10, 8, 6, 7, 1, 4, 2 , 5 } となる．

※提出課題【問題２】と実質同じものである．注目している要素は v に格納し，
最終的な格納場所が決定するまでは動かさない．
upheap() の場合は，順序関係が不成立な場合は親を引きずり下ろすような形になる．
演習で実装したプログラムをよく確認すること．