堆排序的实现。。闲的没事手撸了一个。。

作者：wizardforcel
原文链接：https://ld246.com/article/1506522207668
来源网站：链滴

```
//HeapUtil.heapSort(arr)

class HeapUtil
{
    public static void swap<T>(IList<T> list, int i, int j)
    {
        T tmp = list[i];
        list[i] = list[j];
        list[j] = tmp;
    }

    public static void adjustUp(IList<int> list, int idx)
    {
        int n = idx;
        while(n != 0)
        {
            int parent = (n - 1) / 2;

            if (list[n] < list[parent])
            {
                swap(list, n, parent);
                n = parent;
            }
            else
                break;

        }
    }

    public static void adjustDown(IList<int> list, int idx)
    {
        int n = idx;
        while (n < list.Count / 2)
        {
            int lchild = n * 2 + 1;
            int smallest = n;
            if(list[lchild] < list[smallest])
            {
                smallest = lchild;
            }

            int rchild = lchild + 1;
            if(rchild < list.Count && list[rchild] < list[smallest])
            {
                smallest = rchild;
            }

            if (smallest == n)
                break;
            else
            {
                swap(list, smallest, n);
                n = smallest;
            }
```

```
        }
    }

    public static void heapAdd(IList<int> list, int elem)
    {
        list.Add(elem);
        adjustUp(list, list.Count - 1);
    }

    public static int heapRm(IList<int> list, int idx)
    {
        swap(list, idx, list.Count - 1);
        int res = list[list.Count - 1];
        list.RemoveAt(list.Count - 1);
        adjustDown(list, idx);
        return res;

    }

    public static void heapMake(IList<int> list)
    {
        for(int i = list.Count / 2 - 1; i >= 0; i--)
        {
            adjustDown(list, i);
        }
    }

    public static void heapSort(IList<int> list)
    {
        heapMake(list);

        IList<int> res = new List<int>();

        while(list.Count != 0)
        {
            res.Add(heapRm(list, 0));
        }

        list.Clear();
        foreach(var e in res) list.Add(e);

    }


}
```