

# Practice 1

---

## 1

---

Implement exercise 2.3-7.

Describe a  $\theta(n \lg n)$ -time algorithm that, given a set  $S$  of  $n$  integers and another integer  $x$ , determines whether or not there exist two elements in  $S$  whose sum is exactly  $x$ .

实现 练习2.3-7

描述一个时间复杂度为  $\theta(n \lg n)$  的算法如下，给定一个含  $n$  个整数的集合  $S$  和一个整数  $x$ ，确定  $S$  中是否存在两个元素的总和为  $x$ 。

## 2

---

Implement priority queue.

- **MAXIMUM(S)** -- Return the maximal element;
- **EXTRACT-MAX(S)** -- Extract the maximal element;
- **INCREASE-KEY(S, x, k)** -- Increase the value of an element;
- **INSERT(S, x)** -- Insert a new element.

实现优先队列

- **MAXIMUM(S)** -- 返回最大元素；
- **EXTRACT-MAX(S)** -- 取出最大元素；（返回并从队列中删除）
- **INCREASE-KEY(S, x, k)** -- 增加某一元素的值；
- **INSERT(S, x)** -- 插入一个新元素。

## 3

---

Implement **Quicksort** and answer the following questions.

1. How many comparisons will Quicksort do on a list of  $n$  elements that all have the same value?
2. What are the maximum and minimum number of comparisons will Quicksort do on a list of  $n$  elements, give an instance for maximum and minimum case respectively.

实现 **快速排序** 并回答以下问题。

1. 在一个含有  $n$  个元素且所有元素值相等的列表上执行快速排序将会进行多少次比较？
2. 在一个含有  $n$  个元素的列表上执行快速排序，最大和最小的比较次数分别是多少，分别举一个例子。

## 4

---

Give a **divide and conquer** algorithm for the following problem: you are given two sorted lists of size  $m$  and  $n$ , and are allowed unit time access to the  $i$ th element of each list. Give an  $O(\lg m + \lg n)$  time algorithm for computing the  $k$ th largest element in the union of the two lists. (For simplicity, you can assume that the elements of the two lists are distinct).

给出以下问题的**分治**算法：给定两个有序且大小分别为  $m$  和  $n$  的列表，并且对于每个列表，获取第  $i$  个元素只需要消耗单位时间。请给出一个时间复杂度为  $O(\lg m + \lg n)$  的算法，计算这两个列表的联合中第  $k$  大的元素。（为简单起见，你可以认为这两个列表中的元素是相异的）