

**5-2** (查找一个无序数组) 本题将分析三个算法, 它们在一个包含  $n$  个元素的无序数组  $A$  中查找一个值  $x$ 。

考虑如下的随机策略: 随机挑选  $A$  中的一个下标  $i$ 。如果  $A[i]=x$ , 则终止; 否则, 继续挑选  $A$  中一个新的随机下标。重复随机挑选下标, 直到找到一个下标  $j$ , 使  $A[j]=x$ , 或者直到我们已检查过  $A$  中的每一个元素。注意, 我们每次都是从全部下标的集合中挑选, 于是可能会不止一次地检查某个元素。

- a. 请写出过程 RANDOM-SEARCH 的伪代码来实现上述策略。确保当  $A$  中所有下标都被挑选过时, 你的算法应停止。
- b. 假定恰好有一个下标  $i$  使得  $A[i]=x$ 。在我们找到  $x$  和 RANDOM-SEARCH 结束之前, 必须挑选  $A$  下标的数目期望是多少?
- c. 假设有  $k \geq 1$  个下标  $i$  使得  $A[i]=x$ , 推广你对 (b) 部分的解答。在找到  $x$  或 RANDOM-SEARCH 结束之前, 必须挑选  $A$  的下标的数目期望是多少? 你的答案应该是  $n$  和  $k$  的函数。
- d. 假设没有下标  $i$  使得  $A[i]=x$ 。在检查完  $A$  的所有元素或 RANDOM-SEARCH 结束之前, 我们必须挑选  $A$  的下标的数目期望是多少?

现在考虑一个确定性的线性查找算法, 我们称之为 DETERMINISTIC-SEARCH。具体地说, 这个算法在  $A$  中顺序查找  $x$ , 考虑  $A[1], A[2], A[3], \dots, A[n]$ , 直到找到  $A[i]=x$ , 或者到达数组的末尾。假设输入数组的所有排列都是等可能的。

- e. 假设恰好有一个下标  $i$  使得  $A[i]=x$ 。DETERMINISTIC-SEARCH 平均情形的运行时间是多少? DETERMINISTIC-SEARCH 最坏情形的运行时间又是多少?
- f. 假设有  $k \geq 1$  个下标  $i$  使得  $A[i]=x$ , 推广你对 (e) 部分的解答。DETERMINISTIC-SEARCH 平均情形的运行时间是多少? DETERMINISTIC-SEARCH 最坏情形的运行时间又是多少? 你的答案应是  $n$  与  $k$  的函数。
- g. 假设没有下标  $i$  使得  $A[i]=x$ 。DETERMINISTIC-SEARCH 平均情形的运行时间是多少? DETERMINISTIC-SEARCH 最坏情形的运行时间又是多少?

最后, 考虑一个随机算法 SCRAMBLE-SEARCH, 它先将输入数组随机变换排列, 然后在排列变换后的数组上, 运行上面的确定性线性查找算法。

- h. 设  $k$  是满足  $A[i]=x$  的下标的数目, 请给出在  $k=0$  和  $k=1$  情况下, 算法 SCRAMBLE-SEARCH 最坏情形的运行时间和运行时间期望。推广你的解答以处理  $k \geq 1$  的情况。
- i. 你将会使用 3 种查找算法中的哪一个? 解释你的答案。

习题答案:

a.

```
RANDOM-SEARCH(A, v):  
    B = new array[n]  
    count = 0  
    while(count < n):  
        r = RANDOM(1, n)  
        if A[r] == v:  
            return r  
        if B[r] == false:  
            count += 1  
            B[r] = true  
    return false
```

b. 就是几何分布 $\sim n$

c. 还是几何分布 $\sim n/k$

d. Section 5.4.2, 在每个盒子里至少有一个球前, 要投多少个球.

分成  $n$  个阶段, 每个阶段代表从投中第  $i$  个球后开始, 到投中第  $i+1$  个球为止的区间, 所以

每个阶段可以当做几何分布, 求和为  $\sum_{i=1}^b \frac{b}{b-i+1} = b \sum_{i=1}^b \frac{1}{i} = b(\ln b + O(1))$

所以答案为  $n(\ln n + O(1))$

e. 平均查找时间是  $n/2$ , 最坏查找时间是  $n$

f. 最坏运行时间是  $n-k+1$ . 平均运行时间是  $n/(k+1)$

g. 都是  $n$

h. 跟 DETERMINISTIC-SEARCH 是一样的, 这时候的期望就是平均.

i. 自然是 DETERMINISTIC-SEARCH