Design and Analysis of Algorithms

Tutorial 3: Divide and Conquer Algorithms

童咏昕

北京航空航天大学 计算机学院

yxtong@buaa.edu.cn

问题1

- 给定一列升序排列的数组，数组元素为各不相同的整数。请设计时间复杂度为$O(logn)$的算法找出数组中满足$A[i] = i$的下标i(如果存在的话)，若数组中存在多个下标满足该条件，返回一个结果即可。

问题1–提示

Index-Search(A,s,t)

Input: $\boldsymbol{A}$ is an sorted array of n distinct elements,$\boldsymbol{s}$ and $\boldsymbol{t}$ are the start index and end index of A respectively
Output: The index $\boldsymbol{i}$ in A such that $A[i] = i$
if *s is equal to t* **then**
  if *$A[s]$ is equal to s* **then**
    return s;
  end
  else
    return -1;
  end
end
$m \leftarrow \lfloor \frac{s+t}{2} \rfloor$;
if *$A[m]$ is equal to m* **then**
  return m;//$O(1)$
end
if $A[m] > m$ **then**
  return Index-Search(A, s, t);//$O(\lfloor \frac{n}{2} \rfloor)$
end
else
  return Index-Search$(A, m+1, t)$;//$O(\lceil \frac{n}{2} \rceil)$
end

问题1–提示

- 数组按升序排列且数组元素各不相同，因此有：
  - 如果对某个m, $A[m] > m$, 那么对任意的$i > m$有$\mathrm{A}[i] > i$.
  - 类似地，如果对某个m, $\mathrm{A}[\mathrm{m}] < \mathrm{m}$, 则对任意的$i < m$有$A[i] < i$.
- 无论哪种情况，都可以排除数组中一半数量的元素，而递归地寻找另一半元素中满足要求的解。因此有递推函数

$$\mathrm{T}(n) = T\left(\frac{n}{2}\right) + O(1)$$

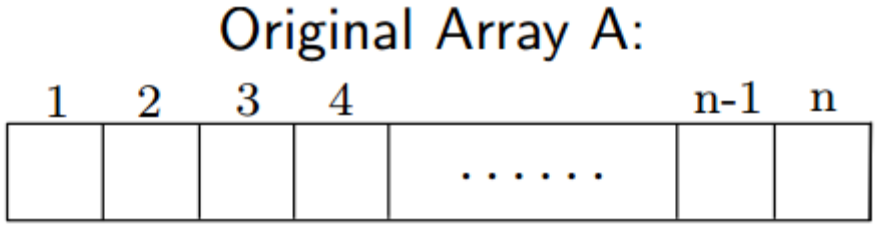
即，$\mathrm{T}(n) = O(\log n)$

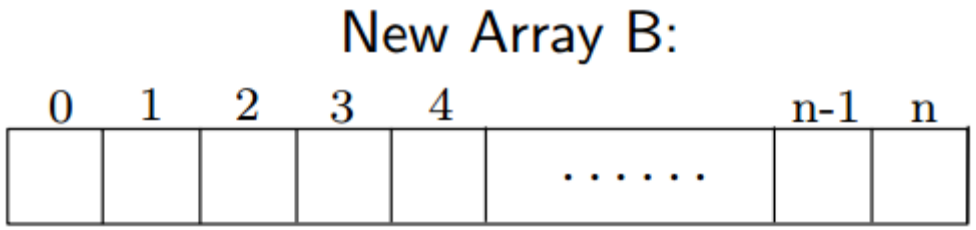
问题2

- 已知算法$\mathcal{A}$能够以$O(n)$的时间复杂度解决下述问题：给定一列整数数组A, 找出数组中使得$A[i] - A[j](i \leq j)$的值最大的下标i, j. 请使用算法$\mathcal{A}$作为子函数, 以$O(n)$的时间复杂度解决最大子数组问题（即：不改变算法$\mathcal{A}$, 而只改变相关算法的输入输出。）

问题2-提示

- 新建数组B[0..n],





令B[0]=0, B[i]=B[i-1]+A[i], 即B[i]为数组A[0..i]的和.

$$B[i] = \sum_{j=1}^{i} A[j], i = 1,2,..,n$$

问题2-提示

- 将数组B作为算法 $\mathcal{A}$ 输入，计算数组B的最大差值。假设算法$\mathcal{A}$找到的数组B中最大值与最小值的下标分别为x和y(y＞x). 最大差值ValAll的值为

$$\begin{aligned} \mathrm{VallAll} &= \mathrm{B[y]} - \mathrm{B[x]} \\ &= \sum_{i=1}^{y} A[i] - \sum_{j=1}^{x} A[j] \\ &= \sum_{i=x+1}^{y} A[i] \end{aligned}$$

因此，ValAll实际上为数组A的连续子数组的和。

ValAll最大化⇒$\sum_{\mathrm{i}=x+1}^{y} A[i]$最大化⇒ValAll为最大子数组问题的结果

问题2–提示

Input-Modifier(A)

Input: $\boldsymbol{A}[1..n]$ is an array of n integers
Output: An array $\boldsymbol{B}[0..n]$
$sum \leftarrow 0$;
$B[0] \leftarrow sum$;
for $i \leftarrow 1$ *to* n **do**
  $sum \leftarrow sum + A[i]$;
  $B[i] \leftarrow sum$;
end
return B;

问题3

- 某公司需要在n个人中聘请1人为秘书，其聘请流程按照Hire-Assistant(n)所示，假设各应聘者之间优于(**better than**)的关系为一个严格全序关系。现假设应聘者按随机顺序到来，求聘用(**hire**)次数的期望。

Hire-Assistant(n)

```
Input: The number of the candidates n
Output: None
best ← 0;
for i ← 1 to n do
    interview candidate i;
    if candidate i is better than best then
        fire best ;
        hire candidate i;
        best ← i;
    end
end
```

问题3–提示

- 令$X_i = 1$表示聘用了应聘者i，$X_i = 0$表示未聘用应聘者i，$X_i = 1$当且仅当应聘者i优于(better than)在他之前出现的应聘者，即前i个应聘者中最优的应聘者出现在位置i，因此有

$$\mathrm{E}[X_i] = \Pr[X_i = 1] = 1/i$$

- 令$X = X_1 + X_2 + \cdots + X_n$表示总的聘用(hire)次数.

$$\begin{aligned}\mathrm{E}[X] &= E[X_1] + E[X_2] + \cdots + E[X_n] \\ &= 1 + \frac{1}{2} + \cdots + \frac{1}{n-1} + \frac{1}{n} \\ &= \Theta(\log n)\end{aligned}$$