

作业14解答:

1. 设计一个 $O(n^2)$ 时间的算法, 找出由 n 个数组成的序列中最长非递减子序列。

1. 解: 设数组 $a[0 \dots n-1]$ 放 n 个给定数字, 数组 $b[0 \dots n-1]$ 表示 $b[i]$ 记录以 $a[i]$ ($0 \leq i < n$) 为结尾元素的最长非递减子序列的长度。 a 中最长非递减子序列的长度为:

$\max_{0 \leq i < n} \{b[i]\}$ 。 $b[i]$ 的递归定义如下:

$$b[0] = 1, \quad b[i] = \max_{\substack{0 \leq k < i \\ a[k] \leq a[i]}} \{b[k]\} + 1。$$

算法如下:

PROCEDURE Longest-Subsequence (VAR a, b : ARRAY $[0 \dots n]$
OF integer);

VAR

i, j, x, k, LS : integer;

BEGIN

$b[0] := 1$;

FOR $i := 1$ TO $n-1$ DO

BEGIN

$x := b[0]$; $k := -1$;

FOR $j := 0$ TO $i-1$ DO

IF $(a[j] \leq a[i])$ AND $(b[j] \geq x)$ THEN

BEGIN

$k := j$;

$x := b[j]$

END;

```

IF k = -1 THEN b[i] := 1
ELSE b[i] := b[k] + 1;
END;
LS := b[0];
FOR i := 1 TO n-1 DO
  IF b[i] > LS THEN LS := b[i];
WRITE ("The length of longest subsequence is", LS);
END;

```

2. 设计算 n 个元素中取 m 个元素的组合数公式如下:

$$\binom{n}{m} = 1, \text{ 如果 } m = 0 \text{ 或 } m = n$$

$$\binom{n}{m} = \binom{n-1}{m} + \binom{n-1}{m-1}, \text{ 如果 } 0 < m < n$$

- 给出计算 $\binom{n}{m}$ 的递归函数;
- 上述递归函数计算的最坏运行时间是什么?
- 写出计算 $\binom{n}{m}$ 的动态规划算法;
- (c) 算法的运行时间是什么?

2. 解: 设计算 $\binom{n}{m}$ 递归调用次数最坏情况下的计算时间为 $T(n)$.

$$(a) \quad T(1) = C_1$$

$$T(n) = 2T(n-1) + C_2$$

(b) 用扩展递推关系式的方法:

$$T(n) = 2T(n-1) + C_2$$

$$= 2(2T(n-2) + C_2) + C_2$$

$$= 2^2 T(n-2) + 2C_2 + C_2$$

$$= 2^2 (2T(n-3) + C_2) + 2C_2 + C_2$$

中山大学本科生考试答题纸

学院(系) _____ 专业 _____ 级 _____

考试科目 _____ 成绩评定 _____

考生姓名 _____ 教师签名 _____

学 号 _____ 年 月 日

警示

《中山大学授予学士学位工作细则》第八条：“考试作弊者不授予学士学位。”

$$= 2^3 T(n-3) + 2^2 C_2 + 2 C_2 + C_2$$

$$= \dots$$

$$= 2^{n-1} T(1) + \left(\sum_{i=0}^{n-2} 2^i \right) C_2$$

$$= 2^{n-1} C_1 + \frac{2^n - 1}{2 - 1} C_2$$

$$\leq C 2^n, \text{ 其中 } C \text{ 为 } \max(C_1, C_2)$$

$$\text{故 } T(n) = O(2^n)$$

(c) 动态规划算法:

PROCEDURE Combin (m, n : integer);

VAR

C : ARRAY [1..max, 0..max] OF integer;

i, j, k : integer;

BEGIN

$C[1, 0] := 1$; $C[1, 1] := 1$;

FOR $i := 2$ TO n DO

BEGIN

$C[i, 0] := 1$; $C[i, i] := 1$;

$k := \text{MIN}(m, i-1)$;

FOR $j := 1$ TO k DO

线
订
装

$C[i, j] := C[i-1, j-1] + C[i-1, j];$

END;

RETURN($C[n, m]$);

END;

(d) 上本算法最外层循环, 循环了 $O(n)$ 次, 内层循环循环了 $O(m)$ 次, 循环内部语句时间为 $O(1)$ 。故整个算法最坏情况下时间复杂度为 $O(mn)$ 。