第2章习题答案

- 第二章

Q15 , Q16

Q18, Q20

Q37(b)(c)(d)(i)

第二章

Q15 , Q16

Q18, Q20

Q37(b)(c)(d)(i)

Q15(n×n矩阵乘法)

The third for loop is entered n^3 times. So, the total number of multiplications is n^3.

Q16(m×n和n×p矩阵乘法)

The third for loop is entered mpn times and the number of multiplications is mpn.

Q18

```
min←0;max←0;
For i←1 to n-1 do
    if a[i]<a[min]  min←i ;
    if a[i]>a[max] max←i.
```

- min←0;max←0;
- For i←1 to n-1 do

  if a[i]<a[min]

  min←i

  else if a[i]>a[max] max←i
- 当输入数组已排好序时,算法要做2(n-1)次比较.当输入为逆序时,算法要做n-1次比较.

Q20

- 程序2.28介绍了一种程序设计技术,即加哨兵方法,以防止数组越界。
- 最好情形做一次关键字比较，最坏情形做 n + 1次比较.
- 和程序2.1相比，多做了一次赋值和比较，少做了n次比较 i < n.

使用步计数法分析下面算法的渐近复杂度

- **Q37(b)**

Statement	s/e	Frequency	Total steps
int MinMax(T a[], int n, int& Min, int& Max)	0	0	$\Theta(0)$
{// Find min and max elements in a[0:n-1].	0	0	$\Theta(0)$
if (n < 1) return 0;	1	1	$\Theta(1)$
Min = Max = 0;	1	1	$\Theta(1)$
for (int i = 1; i < n; i++) {	1	$\Theta(n)$	$\Theta(n)$
if (a[Min] > a[i]) Min = i;	1	$\Theta(n)$	$\Theta(n)$
if (a[Max] < a[i]) Max = i;	1	$\Theta(n)$	$\Theta(n)$
}	0	0	$\Theta(0)$
return 1;	1	1	$\Theta(1)$
}	0	0	$\Theta(0)$

$$t_{MinMax}(n) = \Theta(n)$$

- Q37(c)

```
template <class T>
bool MinMax(T a[], int n, int & Min, int & Max)
{// Locate min and max elements in a[0:n-1].
 // Return false if less than one element.
   if (n < 1) return false;
   Min = Max = 0; // initial guess
   for (int i = 1; i < n; i++)
     if (a[Min] > a[i]) Min = i;
     else if (a[Max] < a[i]) Max = i;
   return true;
}
```

- Q37(d)

Statement	s/e	Frequency	Total steps
void Mult(T **a, T **b, T **c, int n)	0	0	$\Theta(0)$
{// Multiply the n x n matrices a and b to get c.	0	0	$\Theta(0)$
for (int i = 0; i < n; i++)	1	$\Theta(n)$	$\Theta(n)$
for (int j = 0; j < n; j++) {	1	$\Theta(n^2)$	$\Theta(n^2)$
T sum = 0;	1	$\Theta(n^2)$	$\Theta(n^2)$
for (int k = 0; k < n; k++)	1	$\Theta(n^3)$	$\Theta(n^3)$
sum += a[i][k] * b[k][j];	1	$\Theta(n^3)$	$\Theta(n^3)$
c[i][j] = sum;	1	$\Theta(n^2)$	$\Theta(n^2)$
}	0	0	$\Theta(0)$
}	0	0	$\Theta(0)$

$$t_{Mult}(n) = \Theta(n^3)$$

- Q37(i)

Statement	s/e	Frequency	Total steps
void Rank(T a[], int n, int r[])	0	0	$\Theta(0)$
{// Rank the n elements a[0:n-1].	0	0	$\Theta(0)$
for (int i = 0; i < n; i++)	1	$\Theta(n)$	$\Theta(n)$
r[i] = 0;	1	$\Theta(n)$	$\Theta(n)$
for (i = 1; i < n; i++)	1	$\Theta(n)$	$\Theta(n)$
for (int j = 0; j < i; j++)	1	$\Theta(\sum_{i=1}^{n-1} i)$	$\Theta(n^2)$
if (a[j] <= a[i]) r[i]++;	1	$\Theta(n^2)$	$\Theta(n^2)$
else r[j]++;	1	$\Omega(0)$, $O(n^2)$	$\Omega(0)$, $O(n^2)$
}	0	0	$\Theta(0)$

$$t_{Rank}(n) = \Theta(n^2)$$

Homework(2)

- 1.用归纳法证明

$$T(N) \le \begin{cases} 0 & \text{if } N = 1 \\ \underbrace{T(\lceil N/2 \rceil)}_{\text{solve left half}} + \underbrace{T(\lfloor N/2 \rfloor)}_{\text{solve right half}} + \underbrace{cN}_{\text{combine}} & \text{otherwise} \end{cases}$$

$$\Rightarrow T(N) \le cN \lceil \log_2 N \rceil$$

- 2.应用master方法求解T(n)=2T(n/2)+**Θ**($n^{1/2}$)
- 3.展开递归树:$T(n)=T(0)+T(n–1)+cn$,并做渐近分析
- 展开T(n)=T(0.1n)+T(0.9n)+**Θ**(n)的递归树并计算递归树的深度和T(n)的渐近值.
- 14章练习33-(a),(b),(c),(d)

Homework (2)

1、证明

令[x]表示x向上取整

设$2^k < n \le 2^{k+1}$, 则有 [logn]=k+1;又因为

$2^{k-1} < n/2 \le 2^k$,即 $2^{k-1} < [n/2] \le 2^k$,所以[log[n/2]]=k;

所以, [logn]= [log[n/2]]+1

设n_1=n/2向下取整, n_2=[n/2] ;

$$
\begin{aligned}
T(n) &= T(n_1)+T(n_2)+cn \\
&\le cn_1[\log n_1]+cn_2[\log n_2]+cn \\
&\le cn[\log n_2]+cn=cn([\log n_2]+1)=cn[\log n]
\end{aligned}
$$

2.应用master方法求解$T(n)=2T(n/2)+\Theta(n^{1/2})$

$a=b=2$, $n^{\log a}=n$, $f(n)=cn^{1/2}$, ;所以

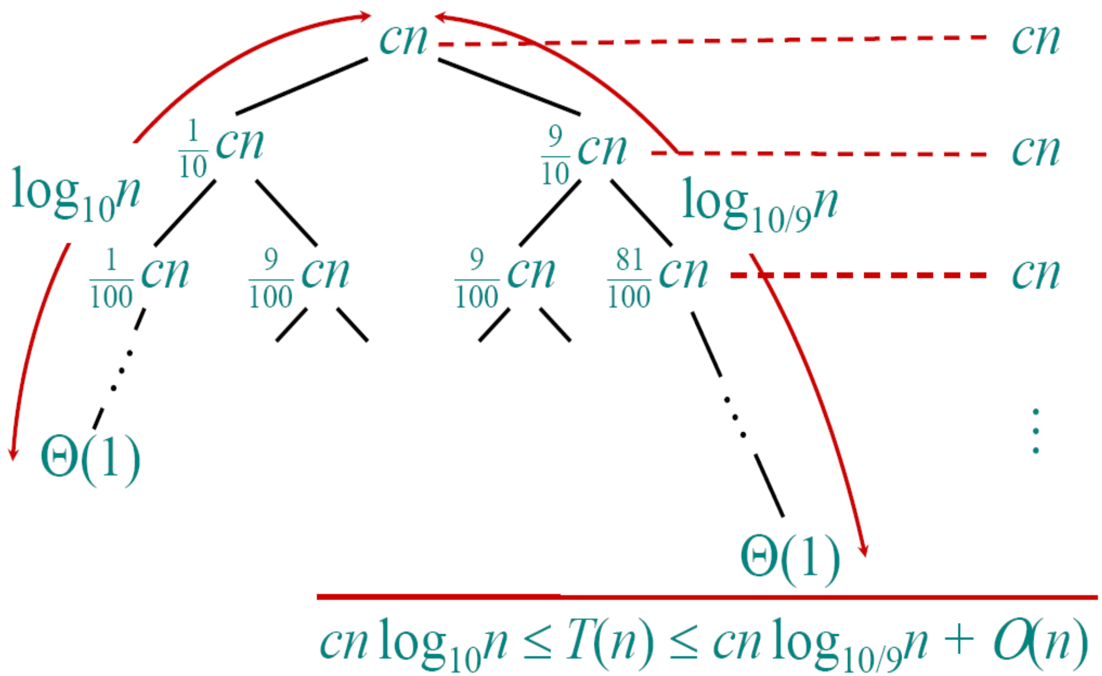
$f(n)=O(n^{1-1/2})$,

$T(n)=\Theta(n)$

3. 展开递归树: $T(n)=T(0)+T(n-1)+cn$, 并做渐近分析

$T(n)=nT(0)+c(1+\ldots+(n-1)+n) = \Theta(n^2)$,

- 展开T(n)=T(0.1n)+T(0.9n)+Θ(n)的递归树并计算递归树的深度和T(n)的渐近值.



练习33

（a）Case 1：$\Theta(n^{\log 10})$

（b）Case 3：$\Theta(n^5)$

（c）Case 2：$\Theta(n^3\log n)$

（d）Case 2：$\Theta(n^3\log^3 n)$

练习33

- (e)case 3:$a=9,b=2$, $f(n)=n^2 2^n$,任取c
  $t(n)=\Theta(n^2 2^n)$
- (f)case 3: $a=3,b=8$, $f(n)=n^2 2^n \log n$,任取c
  $t(n)=\Theta(n^2 2^n \log n)$
- (g)case 1:$a=128,b=2,f(n)=6n$
  $t(n)=\Theta(n^7)$
- (h)case3,$\log_b a=7,f(n)=n^8=\Omega(n^{7+\varepsilon})$,
  $8(n/2)^8<cn^8,t(n)=\Theta(n^8)$
- (i)case 3,$t(n)= \Theta(2^n/n)$
- (j)case 1,$t(n)= \Theta(n^7)$