Week 1

软73 沈冠霖 2017013569

March 7, 2019

1 T1

第一问 设$m = lgn$,则$T(2^m) = 2T(2^{\frac{m}{2}}) + 1$，令$T(2^m) = S(m)$，则有$S(m) = 2S(\frac{m}{2}) + 1$，$\log_2^2 = 1$,存在$\epsilon = 0.5$使得1=O(n-0.5)，因此$S(m) = \theta(m), T(n) = \theta(lgn)$

第二问 T(n)=$\theta(1)$，证明：2T(2)=0T(1)+2,T(2)=1。假设对于$\forall 1 < k < n, \exists c_1, c_2 > 0$,有$c_1 <= T(k) <= c_2$,则$n * c_1 + (2 - 2c_1) <= nT(n) <= n * c_2 + (2 - 2c_2)$,只需要让$c_1 = c_2 = 1$,就有$\exists n_0 = 2, \forall n > n_0$,有$c_1 <= T(n) <= c_2$,因此$T(n) = \theta(1)$

2 T2

注：两个实验报告都写到这里了，运行环境是windows10+vs2017

思路 一共对比了三种计算斐波那契数的方法：递归法,递推法（记录每次数据）,矩阵分治法。对比了数据范围在10,50,500,1000,3000,10000,100000,1000000,10000000,50000000情况下三种算法的时间，以及结果。结果见Table1.

结果分析 首先，三种运算的结果完全一致，说明算法正确。
其次，递归到n=50时候时间就已经达到接近100s，作为指数级别复杂度的算法很慢。 最终，递推和矩阵运算都很快，但是矩阵运算虽然时间复杂度是$\theta(lgn)$,却比时间复杂度$\theta(n)$的递推要慢，因为我把矩阵封装成了类，这样常数就很大，同时，因为递归有各种系统的栈操作，也减慢了速度。

3 T3

思路 一共对比了三种计算矩阵乘法的方法：使用eigen库计算，三次循环计算，递归计算。计算了n = 16,32,64,128的矩阵（再大就无法分配内存），并且比较了计算结果是否一致。结果见Table2。

Table 1: 斐波那契数计算

测量序号	1	2	3	4	5	6	7	8	9	10
数据范围	10	50	500	1000	3000	10^4	10^5	10^6	10^7	$5*10^7$
结果	55	10^{11}	$2*10^{19}$	溢出	溢出	溢出	溢出	溢出	溢出	溢出
第一组(递归)时间 (ms)	0	94316	太大	太大	太大	太大	太大	太大	太大	太大
第二组(递推)时间 (ms)	0	0	0	0	0	0	0	2	25	118
第三组(矩阵分治)时间(ms)	0	0	0	0	0	1	5	51	520	2601

注：因为数据太大，计算结果近似用科学计数法表示，保留一位有效数字

结果分析 首先，Eigen库的速度是最快的。其次，在小数据范围内，因为常数的影响比n影响大很多，再加上递归有栈内存等诸多问题，递归法速度远远慢于循环计算。

Table 2: 矩阵计算

测量序号	1	2	3	4
数据范围	16	32	64	128
结果是否正确	是	是	是	是
第一组(Eigen)时间 (ms)	0	0	0	2
第二组(循环)时间 (ms)	0	0	1	2
第三组(递归)时间(ms)	2	59	1113	26342