数据结构与算法 作业2

2019201409 于倬浩

4-1 Recurrence examples

1. $T(n) = 2T(n/2) + n^4$

   $a = 2, b = 2, log_b a = log_2 2 = 1$，$n^4 = \Omega(n^{1+\epsilon})$

   $T(n) = \Theta(n^4)$

2. $T(n) = T(7n/10) + n$

   $a = 1, b = 10/7, log_b a = log_{10/7} 1 = 0, n = \Omega(n^{0+\epsilon})$

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

3. $T(n) = 16T(n/4) + n^2$

   $a = 16, b = 4, log_b a = log_4 16 = 2, n^2 = \Theta(n^2)$

   $T(n) = \Theta(n^2 lgn)$

4. $T(n) = 7T(n/3) + n^2$

   $a = 7, b = 3, log_b a = log_3 7 < 2, n^2 = \Omega(n^{log_3 7+\epsilon})$

   $T(n) = \Theta(n^2)$

5. $T(n) = 7T(n/2) + n^2$

   $a = 7, b = 2, log_b a = log_2 7 > 2, n^2 = O(n^{log_2 7-\epsilon})$

   $T(n) = \Theta(n^{log_2 7})$

6. $T(n) = 2T(n/4) + \sqrt{n}$

   $a = 2, b = 4, log_b a = log_4 2 = 1/2, \sqrt{n} = \Theta(n^{log_4 2})$

   $T(n) = \Theta(\sqrt{n} lgn)$

7. $T(n) = T(n-2) + n^2$

   $T(n) = T(n-2) + n^2 = T(n-4) + n^2 + (n-2)^2 = \ldots = \sum_{i=1}^{\frac{n}{2}} (2i)^2 = 4\sum_{i=1}^{\frac{n}{2}} (i)^2$

$T(n) = \Theta(n^3)$

4-3 More recurrence examples

1. $T(n) = 4T(n/3) + nlgn$

   $a = 4, b = 3, nlgn = O(n^{log_3 4 - \epsilon})$

   $T(n) = \Theta(n^{log_3 4})$

2. $T(n) = 3T(n/3) + n/lgn$

   $T(n) = 3T(n/3) + n/lgn = 9T(n/9) + n/lg(n/3) + n/lgn$

   $= \ldots = \sum_{i=0} \frac{n}{lg(n/(3^i))} = \sum_{i=0} \frac{n}{log_3(n) - i}$

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

3. $T(n) = 4T(n/2) + n^2\sqrt{n}$

   $a = 4, b = 2, n^2\sqrt{n} = \Omega(n^{log_2 4 + \epsilon})$

   $T(n) = \Theta(n^2\sqrt{n})$

4. $T(n) = 3T(n/3 - 2) + n/2$

   令$m = n - 6, T(m) = 3T(m/3) + m/2$

   $a = 3, b = 3, m/2 = \Theta(m)$

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

5. $T(n) = 2T(n/2) + n/lgn$

   $T(n) = 2T(n/2) + n/lgn = 4T(n/4) + n/lg(n/2) + n/lgn$

   $= \ldots = \sum_{i=0} \frac{n}{lg(n/(2^i))} = \Theta(nlglgn)$

6. $T(n) = T(n/2) + T(n/4) + T(n/8) + n$

   假设已知$T(n) = \Theta(n), T(n) \leq cn$

   $T(n) = T(n/2) + T(n/4) + T(n/8) + n \leq \frac{7}{8}cn + n = (\frac{7}{8}c + 1)n$

   $\forall n, (\frac{7}{8}c + 1)n \leq cn, \; s.t. n \leq \frac{1}{8}cn$

   $\therefore$ 取$c \geq 8, T(n) \leq cn, T(n) = \Theta(n)$

7. $T(n) = T(n-1) + 1/n$

   $T(n) = T(n-1) + 1/n = \sum_{i=1}^{n} \frac{1}{i} = \Theta(lnn)$

8. $T(n) = T(n-1) + lgn$

   $T(n) = T(n-1) + lgn = \sum_{i=1}^{n} lg(i) = lg(\prod_{i=1}^{n}) = \Theta(lg(n!))$

9. $T(n) = T(n-2) + 1/lgn$

   $T(n) = \sum_{i=1}^{\frac{n}{2}} \frac{1}{lg(2i)}$

10. $T(n) = \sqrt{n}T(\sqrt{n}) + n$

    $T(n) = \sqrt{n}(\sqrt{\sqrt{n}}T(\sqrt{\sqrt{n}}) + \sqrt{n}) + n = n^{\frac{3}{4}}T(n^{\frac{1}{4}}) + 2n = n + \sum_{i=1} n^{\frac{2^{i-1}+1}{2^i}}$

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

4-6 Monge arrays

1. 1)若矩阵为Monge Array时，令 k=i+1, l=j+1

   则有A[i,j]+A[k,l] = A[i,j]+A[i+1,j+1] ⩽ A[i,j+1]+A[i+1,j]

   因此当矩阵为Monge Array时，一定满足上式。

   2)若对于一个m*n的矩阵，满足A[i,j]+A[i+1,j+1] ⩽ A[i,j+1]+A[i+1,j]

   任取i,j，有A[i,j]+A[i+1,j+1] ⩽ A[i,j+1]+A[i+1,j]   ·····(1)

   A[i+1,j]+A[i+2,j+1] ⩽ A[i+1,j+1]+A[i+2,j]   ·····(2)

   (1)(2)求和，有A[i,j]+A[i+2,j+1] ⩽ A[i,j+1]+A[i+2,j] ·····(3)

   因此可以任意叠加行数递增的式子，化简得到

   A[i,j]+A[i+k,j+1] ⩽ A[i,j+1]+A[i+k,j]   ·····(4)

   即从相邻两行扩展至任意两行的同列元素。

   同理，将(4)累加，可以得到

   A[i,j]+A[i+k,j+p] ⩽ A[i,j+p]+A[i+k,j]，即原矩阵满足Monge Array性质。

2.

$$
\begin{matrix}
37 & 23 & 24 & 32 \\
21 & 6 & 7 & 10 \\
53 & 34 & 30 & 31 \\
32 & 13 & 9 & 6 \\
43 & 21 & 15 & 8
\end{matrix}
$$

3. 反证法。

   设第i行j列为第i行的最小元素，第i+1行k列为第i+1行的最小元素，且j>k

   则`A[i,j]+A[i+1,k]`⩽`A[i,k]+A[i+1,j]`与Monge Array的定义不符。

   因此假设不成立，任意相邻两行的最小元素下标单调不减，原结论成立。

4. 假设当前正在处理奇数行`i`，则偶数行`i-1`和`i+1`的最小下标均已计算，且`pos[i-1]`⩽`pos[i]`⩽`pos[i+1]`

   因此，只需线性计算第`i`行`[pos[i-1],pos[i+1]]`下标范围内的最小值即可。

   由于计算所有奇数行时，计算最小值的指针是单调移动的，所以计算所有$\frac{m}{2}$个奇数行所需的时间复杂度为$\Theta(m+n)$

5. $T(m)=T(m/2)+m/2+kn$

   $a=1,b=2,log_b a=0,m/2=\Omega(m^{0+\epsilon})$

   忽略T(m)中的kn项，根据主定理可知含m项的复杂度为$\Theta(m)$

   考虑到这样的递归深度为$O(lgm)$，每次有kn的计算量，因此kn项对复杂度的贡献是$\Theta(nlgm)$

   $\therefore T(m)=\Theta(m+nlgm)$