

第 4 讲: 分治法与递归

姓名: 林凡琪 学号: 211240042

评分: _____ 评阅: _____

2022 年 3 月 16 日

请独立完成作业, 不得抄袭。
若得到他人帮助, 请致谢。
若参考了其它资料, 请给出引用。
鼓励讨论, 但需独立书写解题过程。

1 作业 (必做部分)

题目 1 (TC 4.1-5)

Use the following ideas to develop a nonrecursive, linear-time algorithm for the maximum-subarray problem. Start at the left end of the array, and progress toward the right, keeping track of the maximum subarray seen so far. Knowing a maximum subarray of $A[1 \dots j]$, extend the answer to find a maximum subarray ending at index $j+1$ by using the following observation: a maximum subarray of $A[1 \dots j+1]$ is either a maximum subarray of $A[1 \dots j]$ or a subarray $A[i \dots j+1]$, for some $1 \leq i \leq j+1$. Determine a maximum subarray of the form $A[i \dots j+1]$ in constant time based on knowing a maximum subarray ending at index j .

Algorithm 1 maximum subarray

解答:

```
1: procedure FIND( $A[n]$ )                                ▷ 有  $n$  个元素的数组
2:    $sum[0] \leftarrow 0$ 
3:   for  $i \leftarrow 1, n$  do
4:      $sum[i] \leftarrow sum[i-1] + A[i]$ 
5:   end for
6:    $minn = 0$ 
7:    $ans = -\infty$ 
8:   for  $i \leftarrow 1, n$  do
9:      $ans \leftarrow \max(ans, sum[i] - minn)$ 
10:     $minn = \min(minn, sum[i])$ 
11:   end for return  $ans$ 
12: end procedure
```

题目 2 (TC 4.3-3)

解答:

Prove: $T(n) \geq cn \lg n$ holds for $n \geq 1$, which $c > 0$ is a constant.

For $n = 1$, $T(1) = 1 \geq c1 \lg 1 = 0$.

For $n > 1$, assume that $\forall m, (m \in \mathbb{N}^+ \wedge m < n \rightarrow T(m) \geq cm \lg m)$

$$\begin{aligned}
 T(n) &= 2T(\lfloor \frac{n}{2} \rfloor) + n \\
 &\geq 2((c\lfloor \frac{n}{2} \rfloor \lg(\lfloor \frac{n}{2} \rfloor)) + n \\
 &\geq c(n-1) \lg(n-1) - cn \lg 2 + n \\
 &\geq c(n-1) \lg(n-1) - cn + n \\
 &= c(n-1) \lg(n \frac{n-1}{n}) - cn + n \\
 &\geq cn \lg n - c \lg n + cn \lg(\frac{n-1}{n}) - cn + n \\
 &\geq cn \lg n - cn - cn - cn + n \\
 &\geq cn \lg n
 \end{aligned}$$

The constant c satisfies $0 < c \leq \frac{1}{3}$.

Therefore, $T(n) = \Omega(n \lg n)$.

Since $T(n) = \Omega(n \lg n)$ and $T(n) = O(n \lg n)$, $T(n) = \Theta(n \lg n)$.

题目 3 (TC 4.4-5)

解答:

On the recursion-tree, the maximum depth is n , the minimal depth is $\log_2 n + 1$, and the total cost is $n(\frac{3}{2})^i - t_i (t_i \geq 0)$ in the i^{th} depth which satisfies $i \leq \log_2 n$.

We ignore t_i , and let the depth be the maximum depth. We have

$$T(n) = O(\sum_{i=0}^{n-1} n(\frac{3}{2})^i) = O(n(\frac{3}{2})^n)$$

Therefore, $T(n) = O(n(\frac{3}{2})^n)$.

题目 4 (TC 4.5-4)

解答:

On the recursion-tree, the maximum depth is n , the minimal depth is $\log_2 n + 1$, and the total cost is $n(\frac{3}{2})^i - t_i (t_i \geq 0)$ in the i^{th} depth which satisfies $i \leq \log_2 n$.

We ignore t_i , and let the depth be the maximum depth. We have

$$T(n) = O(\sum_{i=0}^{n-1} n(\frac{3}{2})^i) = O(n(\frac{3}{2})^n)$$

Therefore, $T(n) = O(n(\frac{3}{2})^n)$.

题目 5 (TC Problem 4-1)

Give asymptotic upper and lower bounds for $T(n)$ in each of the following recurrences.

Assume that $T(n)$ is constant for $n \leq 2$. Make your bounds as tight as possible, and justify your answers.

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

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

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

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

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

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

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

解答:

(a)

$$\frac{n^4}{n^{\lg_2 2}} = n^3, \quad af\left(\frac{n}{b}\right) = 2\left(\frac{n}{2}\right)^4 = \frac{1}{8}n^4 \leq cf(n). \quad c \text{ is a constant which satisfies } \frac{1}{8} \leq c < 1.$$

The maser method can be used in $T(n)$.

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

(b)

$$\frac{n}{n^{\lg_{10} 1}} = n, \quad af\left(\frac{n}{b}\right) = \frac{7}{10}n \leq cf(n). \quad c \text{ is a constant which satisfies } \frac{10}{7} \leq c < 1.$$

The maser method can be used in $T(n)$.

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

(c)

$$\frac{n^2}{n^{\lg_4 16}} = 1 \rightarrow n^2 = \Theta(n^{\lg_4 16}).$$

The maser method can be used in $T(n)$.

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

(d)

$$\frac{n^3}{n^{\lg_3 7}} = n^{3 - \lg_3 7}, \quad af\left(\frac{n}{b}\right) = 7\left(\frac{n}{3}\right)^2 = \frac{7}{9}n^2 \leq cf(n). \quad c \text{ is a constant which satisfies } \frac{7}{9} \leq c < 1.$$

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

(e)

$$\frac{n^2}{n^{\lg_2 7}} = n^{2 - \lg_2 7}$$

The maser method can be used in $T(n)$.

$$T(n) = \Theta(n^{\lg_2 7})$$

(f)

$$\frac{n^{\frac{1}{2}}}{n^{\lg_4 2}} = 1 \rightarrow \sqrt{n} = \Theta(n^{\lg_4 2})$$

The maser method can be used in $T(n)$.

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

(g)

Assume that n is even, which doesn't affect the result.

$$T(n) = \sum_{i=1}^{\frac{n}{2}} (2i)^2 = \Theta(n^3).$$

$$\text{So } T(n) = \Theta(n^3)$$

题目 6 (TC Problem 4-3 (Except f and j))

More recurrence examples Give asymptotic upper and lower bounds for $T(n)$ in each of the following recurrences. Assume that $T(n)$ is constant for sufficiently small n . Make your bounds as tight as possible, and justify your answers.

a. $T(n) = 4T(n/3) + n \lg n$

b. $T(n) = 3T(n/3) + n/\lg n$

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

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

e. $T(n) = 2T(n/2) + n/\lg n$

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

h. $T(n) = T(n-1) + \lg n$

i. $T(n) = T(n-2) + 1/\lg n$

解答:

(a)

$n \lg n = O(n^{\log_3 4 - \epsilon})$. ϵ is a constant which satisfies $0 < \epsilon < \log_3 4 - 1$.

The maser method can be used in $T(n)$.

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

(b)

On the recursion-tree, The depth is $\log_3 n + 1$, and the total cost is $\frac{n}{\lg n - i \lg 3}$ nodes in the i^{th} depth.

$$\sum_{i=0}^{\log_3 n} \frac{n}{\lg n - i \lg 3} = \Omega\left(\sum_{i=0}^{\log_3 n} \frac{n}{\lg n}\right) = \Omega(n)$$

So $T(n) = \Omega(n)$

Prove: $T(n) \leq cn \lg n$.

$$\begin{aligned} T(n) &= 3T\left(\frac{n}{3}\right) + \frac{n}{\lg n} \\ &= cn \lg n - n \lg 3 + \frac{n}{\lg n} \\ &\leq cn \lg n \end{aligned}$$

We choose $c = 10$ and $\forall n_0 \in \{2, 3, 4, 5\}$, $(T(n_0) \leq cn_0 \lg n_0)$.

So $T(n) = O(n \lg n)$.

(c)

$\frac{n^2 \sqrt{n}}{n^{\log_2 4}} = n^{\frac{1}{2}}$, $af\left(\frac{n}{b}\right) = 4\left(\frac{n}{2}\right)^2 \sqrt{\frac{n}{2}} = \frac{\sqrt{2}}{2} n^2 \sqrt{n} \leq cf(n)$, which c is a constant which satisfies $\frac{\sqrt{2}}{2} \leq c < 1$.

The maser method can be used in $T(n)$.

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

(d)

Prove: $T(n) \geq cn \lg n$

$$\begin{aligned} T(n) &= 3T(n/3 - 2) + \frac{1}{2}n \\ &= 3c(n/3 - 2) \lg(n \frac{n-6}{3n}) + \frac{1}{2}n \\ &= cn \lg n + cn \lg \frac{n-6}{3n} - 6c \lg n - 6c \lg \frac{n-6}{3n} + \frac{1}{2}n \\ &\geq cn \lg n - 2cn - 6c \lg n + 12c + \frac{1}{2}n \\ &\geq cn \lg n \end{aligned}$$

It will fit when $n > 24$ and let $c = \frac{1}{26}$.

So $T(n) = \Omega(n \lg n)$.

For $T_1(n) = 3f(n/3) + \frac{1}{2}n$, we have that $T(n) = O(T_1(n))$.

With the maser method, $T_1(n) = \Theta(n \lg n)$, so $T(n) = O(n \lg n)$.

Since $T(n) = \Omega(n \lg n)$ and $T(n) = O(n \lg n)$, $T(n) = \Theta(n \lg n)$.

(e)

On the recursion-tree, The depth is $\log_2 n + 1$, and the total cost is $\frac{n}{\lg n - i}$ nodes in the i^{th} depth.

$$\sum_{i=0}^{\log_2 n} \frac{n}{\lg n - i} = \Omega\left(\sum_{i=0}^{\log_2 n} \frac{n}{\lg n}\right) = \Omega(n)$$

So $T(n) = \Omega(n)$

Prove: $T(n) \leq cn \lg n$.

$$\begin{aligned} T(n) &= 2T\left(\frac{n}{2}\right) + \frac{n}{\lg n} \\ &= cn \lg n - cn + \frac{n}{\lg n} \\ &\leq cn \lg n \end{aligned}$$

We choose $c = 10$ and $\forall n_0 \in \{2, 3\}$, $(T(n_0) \leq cn_0 \lg n_0)$.

So $T(n) = O(n \lg n)$.

(g)

$$T(n) = \Theta\left(\sum_{i=1}^n \frac{1}{n}\right) = \Theta(H_n).$$

So $T(n) = \Theta(\lg n)$.

(h)

$$T(n) = \Theta\left(\sum_{i=1}^n \lg i\right)$$

$$\Theta\left(\sum_{i=1}^n \lg i\right) = \Theta(\lg n!) = \Theta(n \lg n)$$

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

(i)

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

$$\sum_{i=1}^{\frac{n}{2}} \left(\frac{1}{\lg(2i)}\right) = \Omega\left(\sum_{i=1}^{\frac{n}{2}} \frac{1}{\lg n}\right) = \Omega\left(\frac{n}{\lg n}\right)$$

$$\text{So } T(n) = \Omega\left(\frac{n}{\lg n}\right)$$

$$\sum_{i=1}^{\frac{n}{2}} \left(\frac{1}{\lg(2i)}\right) = O\left(\sum_{i=1}^{\frac{n}{2}} 1\right) = O(n)$$

$$\text{So } T(n) = O(n)$$

2 作业 (选做部分)

题目 1 (TC Problem 4-3 (f and j))

解答:

3 Open Topics

Open Topics 1 (Akra-Bazzi Method)

介绍求解递归式的 Akra-Bazzi Method, 比如定理介绍、应用与简要证明思路。

参考资料:

- 论文 “On the Solution of Linear Recurrence Equations” ^①。
- [Akra-Bazzi method @ wiki](#)
- 更多精彩, 由你掌握。

Open Topics 2 (Merge-Sort)

请你深入分析 MERGE-SORT, 例如:

- 严格求解 MERGE-SORT 的递推式

$$T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + N, \text{ for } n > 1 \text{ with } T(1) = 0.$$

参考资料: Section 2.6 of “An Introduction to the Analysis of Algorithm” (2nd Edition) ^②。

- MERGE 阶段的下界。重点介绍两个有序数组大小相同的情况; 可概述其它情况。
参考资料: Section 5.3.2 “Minimum Comparison Merging” of TAOCPP Vol 3 ^③。

- 更多精彩, 由你掌握。

2-3.1

$C(k + n - 1, n)$

2-3.4

$(a)C(2n, n) * A(n, n)$

4 反馈