第 4 讲: 分治法与递归

姓名： 朱宇博　　**学号：** 191220186

评分： ______　　**评阅：** ______

2020 年 3 月 20 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）

We assume that the root of the recursion-tree is in the 0^{th} depth.

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 $A[1 \ldots j]$, extend the answer to find a maximum subarray ending at index $j+1$ by using the following observation: a maximum subarray of $A[1 \ldots j+1]$ is either a maximum subarray of $A[1 \ldots j]$ or a subarray $A[i \ldots j+1]$, for some $1 \leq i \leq j+1$. Determine a maximum subarray of the form $A[i \ldots j+1]$ in constant time based on knowing a maximum subarray ending at index j.

解答：

Algorithm 1 maximum-subarray

procedure FIND(A[],n) ▷ an array with n elements
　sum[0]$\leftarrow$ 0
　for $i \leftarrow 1$ to n **do**
　　sum[i]$\leftarrow$ sum[$i-1$]+A[i]
　end for
　minn=0
　ans=$-\infty$
　for $i \leftarrow 1$ to n **do**
　　ans$\leftarrow$max(ans,sum[i]-minn)
　　minn$\leftarrow$min(minn,sum[i])
　end for
　return ans
end procedure

题目 2 (TC 4.3-3)
We saw that the solution of $T(n) = 2T(\lfloor \frac{n}{2} \rfloor) + n$ is $O(n \lg n)$. Show that the solution of this recurrence is also $\Omega(n \lg n)$. Conclude that the solution is $\Theta(n \lg n)$

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 $\forall m, (m \in \mathbb{N}^+ \wedge m < n \to T(m) \geq cn \lg n)$
$T(n) = 2T(\lfloor \frac{n}{2} \rfloor) + n$
$\quad \geq 2((c\lfloor \frac{n}{2} \rfloor \lg(\lfloor \frac{n}{2} \rfloor)) + n$
$\quad \geq c(n-1)\lg(n-1) - cnlg2 + n$
$\quad \geq c(n-1)\lg(n-1) - cn + n$
$\quad = c(n-1)\lg(n\frac{n-1}{n}) - cn + n$
$\quad \geq cn \lg n - c \lg n + cn \lg(\frac{n-1}{n}) - cn + n$
$\quad \geq cn \lg n - cn - cn - cn + n$
$\quad \geq cn \lg n$
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)
Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = T(n-1) + T(\frac{n}{2}) + n$. Use the substitution method to verify your answer.

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
$T(n) = O(\sum\limits_{i=0}^{n-1} n(\frac{3}{2})^i) = O(n(\frac{3}{2})^n)$
Therefore, $T(n) = O(n(\frac{3}{2})^n)$.

题目 4 (TC 4.5-4)
Can the master method be applied to the recurrence $T(n) = 4T(n/2) + n^2 \lg n$? Why or why not? Give an asymptotic upper bound for this recurrence.

解答:
The master method can not be used to solve that, since $\frac{n^2 \lg n}{n^{\log_2 4}} = \lg n$ which is not not polynomially larger.
On the recursion-tree, The depth is $log_2 n + 1$, and the total cost is $n^2 log_2 n - n^2 i$ nodes in the i^{th} depth.
$\sum\limits_{i=0}^{\log_2 n} n^2 log_2 n - n^2 i = O(\sum\limits_{i=0}^{\log_2 n} n^2 log_2 n)$
Therefore, $T(n) = O(n^2 log^2 n)$.

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 $n \le 2$. 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)
$\frac{n^4}{n^{\log_2 2}} = n^3$, $af(\frac{n}{b}) = 2(\frac{n}{2})^4 = \frac{1}{8}n^4 \le cf(n)$. c is a constant which satisfies $\frac{1}{8} \le c < 1$.
The maser method can be used in T(n).
$T(n) = \Theta(n^4)$

(b)
$\frac{n}{n^{\log_{\frac{10}{7}} 1}} = n$, $af(\frac{n}{b}) = \frac{7}{10}n \le cf(n)$. c is a constant which satisfies $\frac{10}{7} \le c < 1$.
The maser method can be used in T(n).
$T(n) = \Theta(n)$

(c)
$\frac{n^2}{n^{\log_4 16}} = 1 \rightarrow n^2 = \Theta(n^{\log_4 16})$.
The maser method can be used in T(n).
$T(n) = \Theta(n^2 \lg n)$

(d)
$\frac{n^3}{n^{\log_3 7}} = n^{3-log_3 7}$, $af(\frac{n}{b}) = 7(\frac{n}{3})^2 = \frac{7}{9}n^2 \le cf(n)$. c is a constant which satisfies $\frac{7}{9} \le c < 1$.
$-\epsilon$ $T(n) = \Theta(n^2)$

(e)
$\frac{n^2}{n^{\log_2 7}} = n^{2-\log_2 7}$
The maser method can be used in T(n).
$T(n) = \Theta(n^{log_2 7})$

(f)
$\frac{n^{\frac{1}{2}}}{n^{\log_4 2}} = 1 \rightarrow \sqrt{n} = \Theta n^{log_4 2}$
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.
$T(n) = \sum_{i=1}^{\frac{n}{2}} (2i)^2 = \Theta(n^3)$.
So $T(n) = \Theta(n^3)$

题目 6 (TC Problem 4-3 (Except f and j))
4-3 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.
$\sum\limits_{i=0}^{\log_3 n} \frac{n}{\lg n - i\lg 3} = \Omega(\sum\limits_{i=0}^{\log_3 n} \frac{n}{\lg n}) = \Omega(n)$
So $T(n) = \Omega(n)$
Prove: $T(n) \le cn\lg n$.
$T(n) = 3T(\frac{n}{3}) + \frac{n}{\lg n}$
$\quad = cn\lg n - n\lg 3 + \frac{n}{\lg n}$
$\quad \le cn\lg n$
We choose $c = 10$ and $\forall n_0 \in \{2,3,4,5\}, (T(n_0) \le cn_0\lg n_0)$.
So $T(n) = O(n\lg n)$.

(c)
$\frac{n^2\sqrt{n}}{n^{log_2 4}} = n^{\frac{1}{2}}$, $af(\frac{n}{b}) = 4(\frac{n}{2})^2\sqrt{\frac{n}{2}} = \frac{\sqrt{2}}{2}n^2\sqrt{n} \le cf(n)$, which c is a constant which satisfies $\frac{\sqrt{2}}{2} \le c < 1$.
The maser method can be used in T(n).
$T(n) = \Theta(n^2\sqrt{n})$

(d)
Prove: $T(n) \ge cn\lg n$
$T(n) = 3T(n/3-2) + \frac{1}{2}n$
$\quad = 3c(n/3-2)\lg(n\frac{n-6}{3n}) + \frac{1}{2}n$
$\quad = cn\lg n + cn\lg\frac{n-6}{3n} - 6c\lg n - 6c\lg\frac{n-6}{3n} + \frac{1}{2}n$
$\quad \ge cn\lg n - 2cn - 6c\lg n + 12c + \frac{1}{2}n$
$\quad \ge cn\lg n$
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.

$\sum\limits_{i=0}^{\log_2 n} \frac{n}{\lg n - i} = \Omega(\sum\limits_{i=0}^{\log_2 n} \frac{n}{\lg n}) = \Omega(n)$
So $T(n) = \Omega(n)$
Prove: $T(n) \le cn \lg n$.
$T(n) = 2T(\frac{n}{2}) + \frac{n}{\lg n}$
$\quad = cn \lg n - cn + \frac{n}{\lg n}$
$\quad \le cn \lg n$
We choose $c = 10$ and $\forall n_0 \in \{2, 3\}, (T(n_0) \le cn_0 \lg n_0)$.
So $T(n) = O(n \lg n)$.

(g)
$T(n) = \Theta(\sum\limits_{i=1}^{n} \frac{1}{n}) = \Theta(H_n)$.
So $T(n) = \Theta(\lg n)$.

(h)
$T(n) = \Theta(\sum\limits_{i=1}^{n} \lg i)$
$\Theta(\sum\limits_{i=1}^{n} \lg i) = \Theta(\lg n!) = \Theta(n \lg n)$
$T(n) = \Theta(n \lg n)$

(i)
$T(n) = \Theta(\sum\limits_{i=1}^{\frac{n}{2}} \frac{1}{\lg(2i)})$
$\sum\limits_{i=1}^{\frac{n}{2}} (\frac{1}{\lg(2i)}) = \Omega(\sum\limits_{i=1}^{\frac{n}{2}} \frac{1}{\lg n}) = \Omega(\frac{n}{\lg n})$
So $T(n) = \Omega(\frac{n}{\lg n})$
$\sum\limits_{i=1}^{\frac{n}{2}} (\frac{1}{\lg(2i)}) = O(\sum\limits_{i=1}^{\frac{n}{2}} 1) = O(n)$
So $T(n) = O(n)$

---

2 作业（选做部分）

题目 1 (TC Problem 4-3 (*f* and *j*))
4-3 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.
f. $T(n) = T(n/2) + T(n/4) + T(n/8) + n$
j. $T(n) = \sqrt{n}T(\sqrt{n}) + n$

解答：
(f)
On the recursion-tree, the maxinum depth is $log_2 n+1$, the minimal depth is $log_8 n+1$, and the total cost is $n(\frac{7}{8})^i$ in the i^{th} depth which satisfies $i \le log_8 n$.
Obviously, $T(n) = \Omega(\sum\limits_{i=0}^{log_8 n} n(\frac{7}{8})^i) = \Omega(n)$, $T(n) = O(\sum\limits_{i=0}^{log_2 n} n(\frac{7}{8})^i) = O(n)$.
Therefore, $T(n) = \Theta(n)$.
(j)
On the recursion-tree, the total cost is n in the i^{th} depth.
Since $\lim\limits_{n \to +\infty} n^{\frac{1}{n}} = 1$, $n^{\frac{1}{n}} = n^{(\frac{1}{2})^{\log_2 n}}$, the depth is $\log_2 n + 1$.

$T(n) = \Theta(\sum_{i=0}^{\log_2 n} n) = \Theta(n \lg n)$.

Question:
We note the depth as k, and k satisfies $n^{2^{-k}} < 2$, we have $k > \log_2(\log_2 n)$.
Then $T(n) = \Theta(n \lg\lg n)$.
For the First method, we go through the recursion-tree until $n \to 1$, and the answer is $\Theta(n \lg n)$.
For the second method, we go through the recursion-tree until $n < 2$, assume that T(p) which satisfies $p \to 2$ is a constant, and the answer is $\Theta(n \lg\lg n)$.
Which method is correct?

---

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 。 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 。 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 TAOCP Vol 3 ③ 。 ③
- 更多精彩, 由你掌握。

4 反馈