

Homework 3

PB17000297 罗晏宸

April 13 2020

1 Algorithm 10.1

以下是上三角方程组回代解法的串行算法的形式化描述。

算法 1 SISD 上回代求解上三角方程组算法

输入: $A_{n \times n}, \mathbf{b} = (b_1, \dots, b_n)^\top$

输出: $\mathbf{x} = (x_1, \dots, x_n)^\top$

```
1: Begin
2:   for  $i = n$  downto 1 do
3:      $x_i = b_i / a_{ii}$ 
4:     for  $j = 1$  to  $i - 1$  do
5:        $b_j = b_j - a_{ji}x_i$ 
6:        $a_{ji} = 0$ 
7:     end for
8:   end for
9: End
```

- 1 请指出串行算法哪些部分可以并行化。
- 2 写出并行算法的形式化描述（需要注明计算模型类型），分析你的算法的时间复杂度

解

- 1 算法第4行的 j 循环可以并行化。

2 在 UMA 模型上回代求解上三角方程组的并行算法如算法2所示。由2和4两行循环可知复杂度为 $p(n) = p$, $t(n) = O(n)$

算法 2 UMA 上回代求解上三角方程组算法

输入: $A_{n \times n}, b = (b_1, \dots, b_n)^T$

输出: $x = (x_1, \dots, x_n)^T$

```

1: Begin
2:   for  $i = n$  downto 1 do
3:      $x_i = b_i / a_{ii}$ 
4:     for all  $P_j, 1 \leq j \leq p$  do
5:       for  $k = j$  to  $i - 1$  step  $p$  do                                ▷  $p$  是处理器数
6:          $b_k = b_k - a_{ki}x_i$ 
7:          $a_{ki} = 0$ 
8:       end for
9:     end for
10:  end for
11: End

```

2 Exercise 7.10

顶点倒塌法是非常有名的求图的连通分量的算法，其基本思想是：连通的相邻的顶点可以合并成一个超顶点，并以它们中最小标号者标记之；此过程可继续在已合并的超顶点之间进行。在下列的算法中。 $C(i)$ 表示与 i 相邻的最小的超顶点号码； $D(i)$ 表示顶点 i 所属连通分量的最小标号的顶点； $C(i) = \min_j \{D(j) | A(i,j)=1, D(i) \neq D(j)\}$ 语句为每个顶点 i 找与它不属于相同分量的相邻的最小号码的顶点 j ；语句 $C(i) = \min_j \{C(j) | D(j)=i, C(j) \neq i\}$ 表示把每个超顶点的根连到最小号码的相邻的超顶点的根上。Hirschberg 的求连通分量算法如下。

算法 3 PRAM-CREW 上 Hirschberg 求连通分量算法

输入: 邻接矩阵 $A_{n \times n}$

输出: 向量 $D(0 : n - 1)$, 其中 $D(i)$ 表示向量 D 的分量

```

1: Begin
2:   for all  $i : 0 \leq i \leq n - 1$  par-do                                ▷ 初始化 (1)
3:      $D(i) = i$ 

```

```

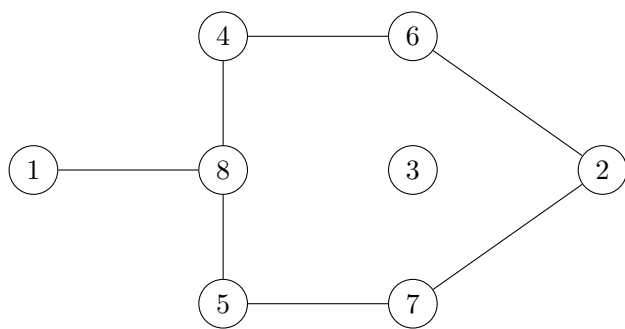
4:   end for
5:   for  $\lceil \log n \rceil$  iterations do
6:       for all  $i, j : 0 \leq i, j \leq n - 1$  par-do ▷ 找相邻的最小者 (2)
7:            $C(i) = \min_j \{D(j) | A(i,j)=1, D(i) \neq D(j)\}$ 
8:           if none then
9:                $C(i) = D(i)$ 
10:          end if
11:      end for
12:      for all  $i, j : 0 \leq i, j \leq n - 1$  par-do ▷ 找每个超顶点的最小相邻超顶点 (3)
13:           $C(i) = \min_j \{C(j) | D(j)=i, C(j) \neq i\}$ 
14:          if none then
15:               $C(i) = D(i)$ 
16:          end if
17:      end for
18:      for all  $i : 0 \leq i \leq n - 1$  par-do ▷ (4)
19:           $D(i) = C(i)$ 
20:      end for
21:      for  $\lceil \log n \rceil$  iterations do ▷ 指针跳跃，找各顶点新的超顶点 (5)
22:          for all  $i : 0 \leq i \leq n - 1$  par-do
23:               $C(i) = C(C(i))$ 
24:          end for
25:      end for
26:      for all  $i : 0 \leq i \leq n - 1$  par-do ▷ (6)
27:           $D(i) = \min \{C(i), D(C(i))\}$ 
28:      end for
29:  end for
30: End

```

(1) 试分析算法3的复杂度 $t(n)$ 和 $p(n)$ 。

(2) 给定如图1所示无向图，试用算法3逐步求出该图的连通分量

解



顶点	1	2	3	4	5	6	7	8
分量	1	2	3	4	5	6	7	8

图 1: 待求连通分量的无向图

(1) 注意到第5行最外侧的循环迭代 $\lceil \log n \rceil$ 次，在循环内第21行循环同样迭代 $\lceil \log n \rceil$ 次，同时考虑外侧循环内的前两个 Do in parallel 循环语句中的求出最小值也是并行的，需要的处理器数为 $(n-1)^2$ ，因此算法的复杂度为 $t(n) = O((\log n)^2)$ ， $p(n) = O(n^2)$ 。

(2) 无向图共有 8 个结点，因此算法共进行 $\lceil \log 8 \rceil = 3$ 次迭代，迭代过程如图2所示。

迭代	顶点 i	1	2	3	4	5	6	7	8
(1)	$D(i)$	1	2	3	4	5	6	7	8
(2)	$C(i)$	8	6	3	6	7	2	2	1
(3)	$C(i)$	8	6	3	6	7	2	2	1
(4)	$D(i)$	8	6	3	6	7	2	2	1
(5)	$C(i)$	1	2	3	2	2	6	6	8
(6)	$D(i)$	1	2	3	2	2	2	2	1
(2)	$C(i)$	1	2	3	1	1	2	2	2
(3)	$C(i)$	2	1	3	2	2	2	2	1
(4)	$D(i)$	2	1	3	2	2	2	2	1
(5)	$C(i)$	1	2	3	1	1	1	1	2
(6)	$D(i)$	1	1	3	1	1	1	1	1
(2)	$C(i)$	1	1	3	1	1	1	1	1
(3)	$C(i)$	1	1	3	1	1	1	1	1
(4)	$D(i)$	1	1	3	1	1	1	1	1
(5)	$C(i)$	1	1	3	1	1	1	1	1
(6)	$D(i)$	1	1	3	1	1	1	1	1

图 2: 算法迭代过程