作业三

李司政 PB19111739

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10.1）

输入：A_{n*n} $b=(b_1, \ldots, b_n)^T$ 　输出：$x=(x_1, \ldots, x_n)^T$

```
Begin
  (1)for i=n downto 1 do
    (1.1)x_i=b_i/a_ii
    (1.2)for j=1 to i-1 do
        b_j=b_j-a_ji x_i
        a_ji=0
      endfor
    endfor
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。

1.第四行的j循环可以并行化

2.

```
SIMD-CREW上回代求解上三角方程组算法
Begin
    for i=n doEndwnto 1 do
        xi=bi/aii
        for all pj，where 1≤j≤p do
            for k=j to i-1 step p do
                bj=bj-ajixi
                aki=0
            end for
        end for
    end for
End
```

j的范围最多从1到n，所有需要的处理器个数为n，并且因为j循环并行化，所有整个j循环的时间复杂度为O(1)，所以$p(n)=O(n)$，$t(n)=O(n)$

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；$C(i)=\min\limits_{j}\{D(j)\mid_{A(i,j)=1,D(i)\neq D(j)}\}$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 $C(i)=\min\limits_{j}\{C(j)\mid_{D(j)=i,C(j)\neq i}\}$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 7.12　PRAM-CREW 上 Hirschberg 求连通分量算法

输入：邻接矩阵 $\boldsymbol{A}_{n\times n}$

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

Begin

(1) **for all** $i: 0\leqslant i\leqslant n-1$ **par-do**　　　　/ * 初始化 * /

$D(i)=i$

endfor

do step (2) through (6) for $\lceil \log n\rceil$ iterations:

(2) **for all** $i,j: 0\leqslant i,j\leqslant n-1$ **par-do**　　　　/ * 找相邻的最小者 * /

(2.1) $C(i)=\min\limits_{j}\{D(j)\mid_{A(i,j)=1 \text{ and } D(i)\neq D(j)}\}$

(2.2) **if** none **then** $C(i)=D(i)$ **endif**

endfor

(3) **for all** $i,j: 0\leqslant i,j\leqslant n-1$ **par-do**　　　　/ * 找每个超顶点的最小相邻超顶点 * /

(3.1) $C(i)=\min\limits_{j}\{C(j)\mid_{D(j)=i \text{ and } C(j)\neq i}\}$

(3.2) **if** none **then** $C(i)=D(i)$ **endif**

endfor

(4) **for all** $i: 0\leqslant i\leqslant n-1$ **par-do** $D(i)=C(i)$ **endfor**

(5) **for** $\lceil \log n\rceil$ iterations **do**　　　　/ * 指针跳越，找各顶点新的超顶点 * /

for all $i: 0\leqslant i\leqslant n-1$ **par-do** $C(i)=C(C(i))$ **endfor**

endfor

(6) **for all** $i: 0\leqslant i\leqslant n-1$ **par-do**

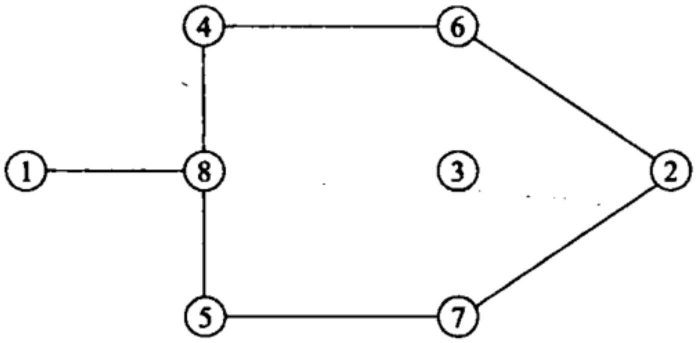
$D(i)=\min\{C(i),D(C(i))\}$

endfor

End

(1) 试分析算法 7.12 的复杂度 $t(n)$ 和 $p(n)$。

(2) 给定如图 7.11 所示无向图，试用算法 7.12 逐步求出该图的连通分量。



顶点	1	2	3	4	5	6	7	8
分量	1	2	3	4	5	6	7	8

图 7.11　待求连通分量的无向图

(1)

do step (2) through (6) for $\lceil \log n \rceil$ iterations：

及

(5) **for** $\lceil \log n \rceil$ iterations **do**　　　　/ * 指针跳越，找各顶点新的超顶点 * /

for all i：$0 \leqslant i \leqslant n-1$ **par-do** $C(i)=C(C(i))$ **endfor**

endfor

两个循环均循环$\lceil logn \rceil$次，且外侧内侧循环均是并行，最多同时需要处理所有的(i,j)对，所以需要的处理器数为n^2个，所以$p(n)=O(n^2)$，$t(n)=O((logn)^2)$

(2)

代码段	顶点	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