

Lab 2 实验报告

2021 年 12 月 6 日

1 问题描述

在 Lab 0 中，我们利用递归方式回溯 LCS 所有结果的输出，但输出过程中存在重复，为了去重我们采用了 set 数据结构，这是一种计算开销很大的算法。例如如下情况， $A = X1234567890$ 和 $B = Xabcdef$ 只有一个 LCS，但是按照朴素算法我们将在每一个节点都进行分岔搜索。面对比较坏的情况，我们将无端多出指数级的开销。

	X	1	2	3	4	5	6	7	8	9	0
X	1	1	1	1	1	1	1	1	1	1	1
a	1	1	1	1	1	1	1	1	1	1	1
b	1	1	1	1	1	1	1	1	1	1	1
c	1	1	1	1	1	1	1	1	1	1	1
d	1	1	1	1	1	1	1	1	1	1	1
e	1	1	1	1	1	1	1	1	1	1	1
f	1	1	1	1	1	1	1	1	1	1	1

图 1: 特殊情况

为了解决冗余输出，一种思路是将 LCS 输出的树结构转化为图结构，将相同节点合并。幸运的是，Ronald I. Greenberg 发明了一种能在 $O(mn)$ 时间开销内将 LCS 二维表压缩为 LCS 图（一种可以表示所有互不相同的 LCS 的图）的算法，本次实验将基于此算法在图结构上遍历打印所有 LCS 输出。

2 算法设计

对任务一，串行打印 LCS Graph 节点，只需唯一且完备遍历所有节点，每次访问时输出即可。常见的解法有 BFS 和 DFS，以 BFS 算法为例，关于 lcsNode 的定义以及 Queue 相关算法的具体实现详见 utils，以下仅展示核心思路。

Algorithm 1 BFS

```
Input graph  $G$ , source node  $s$ 
Let  $Q$  be queue
 $Q.enqueue(s)$ 
mark  $s$  as visited
while  $Q$  is not empty do
   $v = Q.dequeue()$ 
  for each neighbour  $w$  of  $v$  in graph  $G$  do
    if  $w$  is not visited then
       $Q.enqueue(w)$ 
      mark  $w$  as visited
    end if
  end for
end while
```

对任务二，串行输出所有 LCS 结果，只需按 DFS 顺序递归调用输出，核心思路如下。

Algorithm 2 DFS by recursion

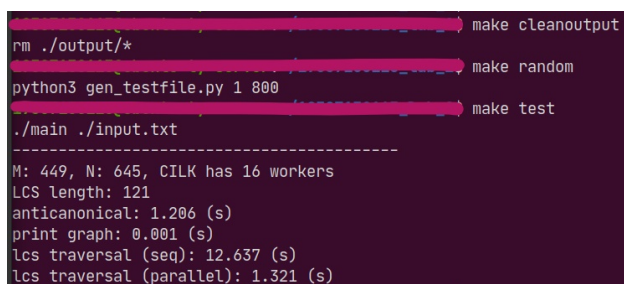
```
Input file pointer  $fp$ , LCS output  $lcs$ , current cell  $cell$ 
if  $cell \rightarrow successors$  is not NULL then
  if  $cell \rightarrow match$  is not NULL then
     $lcs[cell \rightarrow rank - 1] = cell \rightarrow match$ 
  end if
  for each successor  $w$  of  $cell$  do
    DFS(  $fp, lcs, w$  )
  end for
else
  print  $lcs$  to  $fp$ 
end if
```

对任务三，要求并行输出所有 LCS 结果。受到 TLS 启示，每个线程使用各自独占的缓存并输

出到各自对应的文件，算法设计如下。假设有 P 个线程，先用如任务一中的 BFS 串行找到至少 P 个分岔，然后将每一个分岔分给线程并且并行执行（线程执行任务是串行执行），如果还有剩余的分岔则再继续展开，重复上面的过程，直至队列清空或者所有分岔被分完。而当线程切换到新任务时，可利用双向链表回溯该节点的父节点，以得到此前串行走过的 LCS 路径。

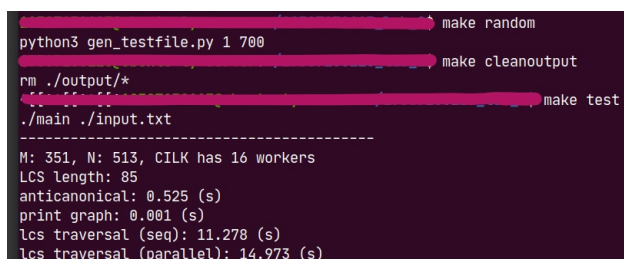
3 实验结果

测试输出如下，正常情况下并行输出速度可以达到串行 10 倍左右；而病态例子（存在大量分岔）下会出现并行速度慢于串行的情况，可能原因是例子阶数不够大，导致并行文件读写带来的开销大于并行加速的增益。



```
rm ./output/*
python3 gen_testfile.py 1 800
./main ./input.txt
-----
M: 449, N: 645, CILK has 16 workers
LCS length: 121
anticanonical: 1.206 (s)
print graph: 0.001 (s)
lcs traversal (seq): 12.637 (s)
lcs traversal (parallel): 1.321 (s)
```

图 2: 实验结果-正常



```
python3 gen_testfile.py 1 700
rm ./output/*
./main ./input.txt
-----
M: 351, N: 513, CILK has 16 workers
LCS length: 85
anticanonical: 0.525 (s)
print graph: 0.001 (s)
lcs traversal (seq): 11.278 (s)
lcs traversal (parallel): 14.973 (s)
```

图 3: 实验结果-反常