EE 382V: Parallel Algorithms **Summer 2017**

Lecture 22: July 23

Lecturer: Vijay Garg *Scribe: Brice Ngnigha*

22.1 Introduction

In this lecture, we will build on our previous lecture on the Euler Tour technique to solve problems such as rooting a tree, computing a vertex level, and computing the number of descendants of a tree.

22.2 Rooting a Tree

Given a tree T = (V, E) rooted at a vertex r. For every vertex v $\neq$ r, we want to find the parent of v, P(v) when the tree T is rooted at vertex r.

Considering the trees below, we compute the root of each vertex:

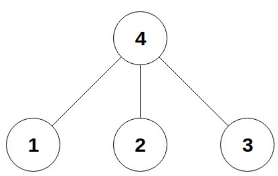


Figure 22.1: Tree rooted at 4

$P(1) = 4 \quad P(2) = 4 \quad P(3) = 4 \quad P(1) = \emptyset$

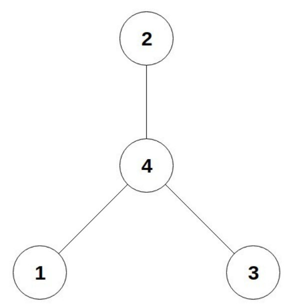


Figure 22.2: Tree rooted at 2

$P(1) = 4 \quad P(1) = 4 \quad P(3) = 4 \quad P(2) = \emptyset$

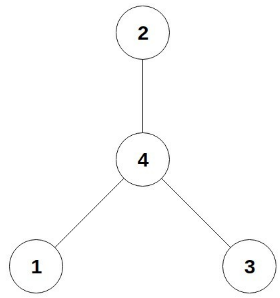
22.2.1 Algorithm

Input : A tree T(Given as adjacency list), a vertex r, the root of T.
Output : For every vertex v $\neq$ r, the parent P(v) of v.

1. Convert the Euler Circuit into the Euler Path
2. Set $succ(u_{d-1}, r)$ to *null*
3. Assign weight w = 1 to each edge in the linked list
   Apply parallel prefix sum on the linked list
4. For each edge (x, y) in parallel do
   if prefsum(x,y) ¡ prefsum(y,x) then P(y) = x

22.2.2 Example

Tree rooted at vertex 2:



$$\text{Euler Circuit}: (1,4) \rightarrow (4,2) \rightarrow (2,4) \rightarrow (4,3) \rightarrow (3,4) \rightarrow (4,1) \rightarrow\rightarrow\rightarrow\rightarrow (1,4)$$

Operation	Result
Euler Path	$(2,4) \rightarrow (4,3) \rightarrow (3,4) \rightarrow (4,1) \rightarrow (1,4) \rightarrow (4,2)$
Weight w	$(1) \rightarrow (1) \rightarrow (1) \rightarrow (1) \rightarrow (1) \rightarrow (1)$
PrefixSum	$(1) \rightarrow (2) \rightarrow (3) \rightarrow (4) \rightarrow (5) \rightarrow (6)$

22.3 Computing the Vertex Level

Given a tree T = (V, E) rooted at a vertex r. For every vertex v $\neq$ r, we want to find the level of vertex v, which is the number of edges from vertex v to the root r.

Considering the tree below, we compute the level of each vertex:

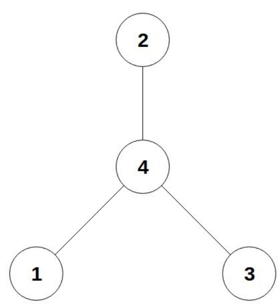


Figure 22.3: Tree rooted at 1
$level(2) = 0$ $level(4) = 1$ $level(1) =$
2 $level(3) = 2$

22.3.1 Algorithm

Input : A tree T(Given as adjacency list), a vertex r, the root of T. *Input* : For every vertex v $\neq$ r,output level(v), the distance from vertex v to the root.

1. Compute Euler Path for the tree T rooted at vertex r
2. For all v r do in par
   w(p(v), v) = -1 ; w(v, p(v)) = -1
3. Perform parallel prefix sum
4. For each v r do in par
   level(v) = prefix sum (p(v), v)
5. level(r) = 0

22.3.2 Example

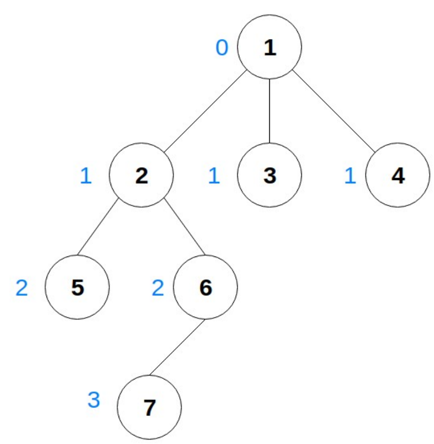


Figure 22.4: Tree rooted at 1

Euler Peth	Weights	Prefix Sum
(1,2)	+1	1
(2,5)	+1	2
(5,2)	-1	1
(2,6)	+1	2
(6,7)	+1	3
(7,6)	-1	2
(6,2)	-1	1
(2,1)	-1	0
(1,3)	+1	1
(3,1)	-1	0
(1,4)	+1	1
(4,1)	-1	0

Figure 22.5: some cap

22.4 Computing the Number of Descendants

Given a tree T = (V, E) rooted at a vertex v. For every vertex v $\neq$ r, we want to find the number of descendants of v, including v.

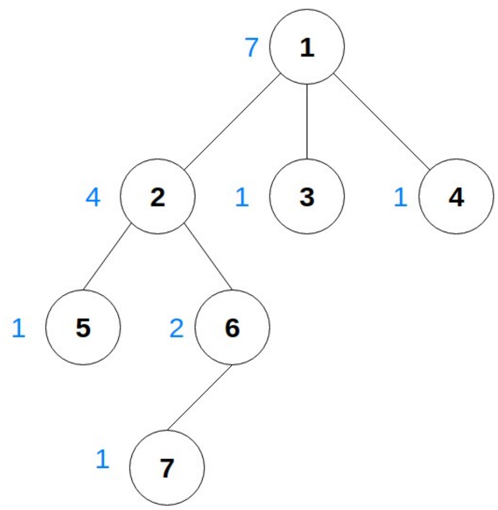
22.4.1 Algorithm

Input : A tree T(Given as adjacency list), a vertex r, the root of T.
Output : For every vertex v $\neq$ r, the number of descendants of vertex v including v.

The Implementation of this algorithm was left as an exercise.
Hint : Size of subtree = total of vertices visited when vertex v is seen last less the number of vertex visited when vertex v is first.

22.4.2 Example

Tree rooted at vertex 1:



The number of descendants is marked in blue.