Techie Delight </>

Types of edges involved in DFS and relation between them

This post describes the types of edges involved in **Depth–first search (DFS)** of a tree and directed & undirected graphs and establish the relation between them.

Prerequisite:

Arrival and departure time of vertices in DFS

Depth–first search in a tree

Depth–first search is a simple **preorder** or **postorder traversal** for a tree, and it contains only tree edges. If `x` is a descendant of `y`, then the relation between the arrival and departure time for tree edges of DFS is:

```
arrival[y] < arrival[x] < departure[x] < departure[y]
```

Depth–first search in an undirected graph

With the graph version of DFS, only some edges will be traversed, and these edges will form a tree, called the **Depth–first search (DFS)** tree of the graph starting at the given root, and the edges in this tree are called **Tree Edges**. One other type of edge called **back edge** points from a node to one of its ancestors in the DFS tree.

For an edge `u —> v` in an undirected graph, the relation between the arrival and departure time for tree edges and back edges is:

Tree edge:

```
arrival[u] < arrival[v]
departure[u] > departure[v]
```

Back edge:

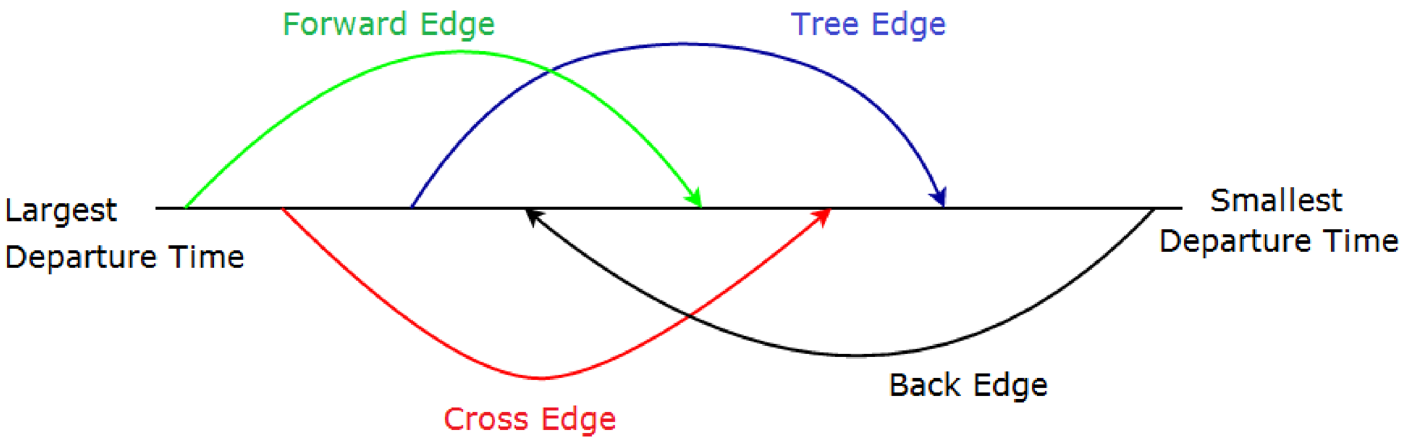
```
arrival[u] > arrival[v]
departure[u] < departure[v]
```

The code for finding arrival and departure time in an undirected graph can be seen **here**.

Depth–first search in a directed graph

There are two other categories of edges of the graph that can be found while doing DFS in a directed graph:

Forward edges that points from a node to one of its descendants.
Cross edges that points from a node to a previously visited node that is neither an ancestor nor a descendant.



For an edge `u –> v` in a directed graph, an edge is a tree edge if `parent[v] = u`. For the other types of edges, we can use their arrival and departure times to tell whether `v` is an ancestor, descendant, or distant cousin of `u`. Following is the relation between the arrival and departure time for different types of edges involved in a DFS of the directed graph:

Tree edge:

```
arrival[u] < arrival[v]
departure[u] > departure[v]
```

Back edge:

```
arrival[u] > arrival[v]
departure[u] < departure[v]
```

Forward edge:

```
arrival[u] < arrival[v]
departure[u] > departure[v]
```

Cross edge:

```
arrival[u] > arrival[v]
departure[u] > departure[v]
```

For tree edge, back edge, and forward edges, the relation between the arrival and departure times of the endpoints is immediate from the tree structure. For any cross edge, `u` is neither an ancestor nor descendant of `v`, So we can say that `u` and `v` intervals does not overlap, i.e., for an edge `u —> v`,

```
arrival[v] < departure[v] < arrival[u] < departure[u]
```

Please note we cannot have an edge from `v —> u`. If any such edge were there, it would have formed a Tree Edge.

References: http://www.cs.yale.edu/homes/aspnes/pinewiki/DepthFirstSearch.html

Graph

Beginner, Depth-first search, Must Know

Techie Delight </>

Resources

All Problems

DSA Practice

Top 100 Most Liked Problems

Top 50 Classic Problems

Top Algorithms

Online IDE

C/C++ Compiler

Java Compiler

Python Compiler

JavaScript Compiler

PHP Compiler

Company

Contact us

Privacy Policy

Terms of Service

Subscribe to new posts