## Introduction

**Heavy-Light Decomposition(HLD)** is a very powerful technique when it comes to solving problems that involve **queries on trees**. The queries on trees may involve finding the sum of the nodes, the maximal/minimal element along the path connecting a given pair of nodes, or updating a set of nodes belonging to a path, and many more.

## Heavy-Light Decomposition

Let us say we are given an array of size N where we need to support two types of queries:
**Type 1:** Update the value of the array at the given **index**. (**1 <= index <= N**)
**Type 2:** Finding the maximum element in the range **L** to **R** (**1 <= L,R <= N, L <= R**).

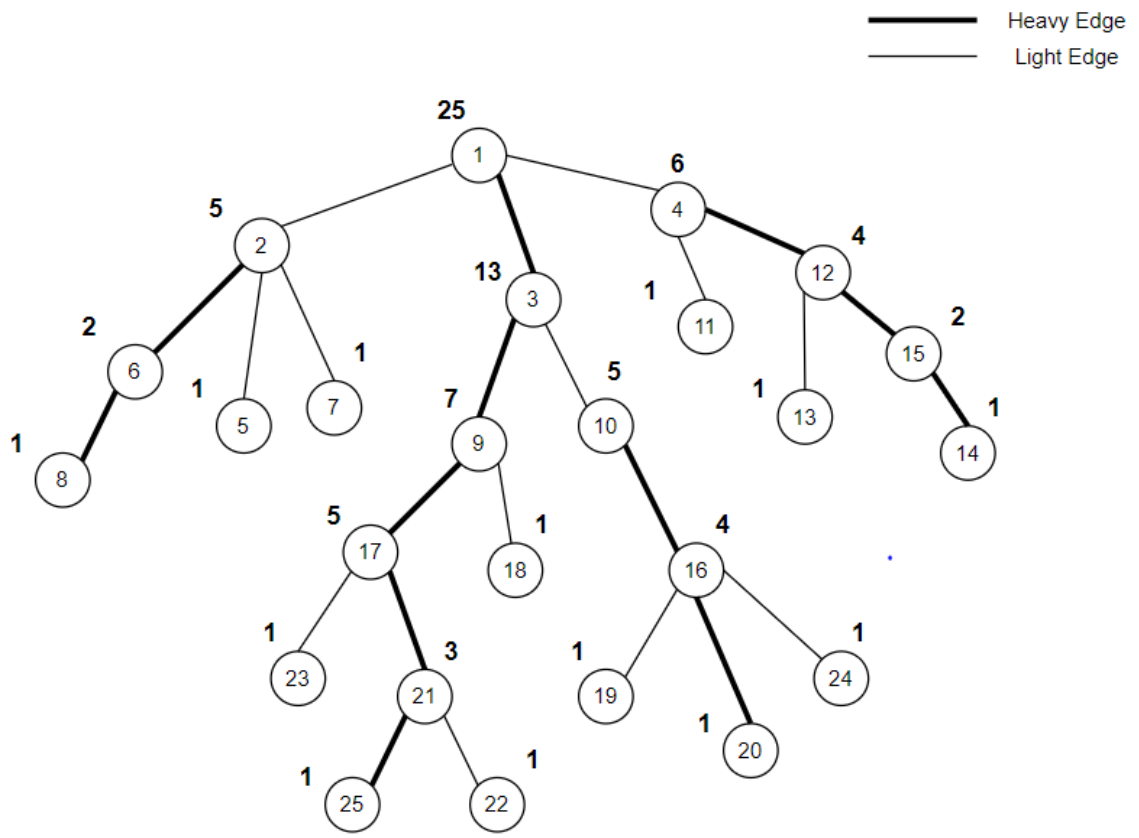
We know that there are various methods and data structures to solve the above problem efficiently like using segment trees, sqrt decomposition, etc.

What about if we need to support both types of queries on a tree?
**Type 1:** Update the value of the given node.
**Type 2:** Finding the maximum value on the path between a given pair of nodes.

A tree has a different structure as compared to an array. One of the ideas can be to **"linearize"** the tree i.e to represent our tree into an array and then apply the known techniques to support the above-mentioned queries and this brings heavy-light decomposition into the picture, which allows us to represent our tree into an array.

Let us introduce some terms that are used in Heavy-Light Decomposition which will become more clear as we go along the lecture:

- **Heavy Child:** A heavy child of a node is the child with the largest subtree size rooted at this child.
- **Light Child:** A child of a node that is not heavy is a light child.
- **Heavy Edge:** An edge that connects a node to its heavy child.
- **Light Edge:** An edge that connects a node to any of its light children.
- **Heavy Path:** A path formed by a collection of heavy edges.
- **Light Path:** A path formed by a collection of light edges.




Let us try to understand each of the terms and definitions mentioned using an example. Consider the tree in the given diagram rooted at **'Node 1'** where the value present inside the circle represents the **label** of the node and the value present above each such node represents the **subtree size rooted at that node**.

**Heavy Child:** As mentioned above a heavy child of a node is the child with the largest subtree size(in case of multiple choices select any arbitrary node), we observe that Node 1 has three children - Node 2, Node 3, and Node 4 where Node 3 is having the maximum subtree size of 13, so Node 1 has Node 3 as "Heavy Child".
A node will always have a single heavy child.
Similarly, the heavy child of different nodes are:

| Node | Heavy Child |
|---|---|
| 2 | 6 |
| 6 | 1 |
| 4 | 12 |
| 3 | 9 |
| 16 | 20 or 19 or 24 |

.... and so on.

**Light Child:** The children of a node except the heavy child are the light children of that node: like Node 1 has Node 2 and Node 3 as light children, Node 16 has Node 19 and Node 24 as light children if Node 20 is considered as the heavy child.

**Heavy Edge:** An edge that connects a node to its heavy child like the edge connecting Node 1 and Node 3 is a heavy edge. All the edges marked as 'bold' are heavy edges.

**Light Edge:** The edge connecting a node to any of its light child, like the edge connecting Node 9 and Node 18, the edge connecting Node 12 and Node 13 are light edges. The edges marked as 'light' are light edges.

**Heavy Path:** The path consisting a collection of heavy edges, like the path connecting Node 1 and Node 25 is a heavy path i.e 1 -> 3 -> 9 -> 17 -> 21 -> 25

**Light Path:** The path consisting of a collection of light edges, like the path connecting Node 1 and Node 5 i.e 1 -> 2 -> 5.