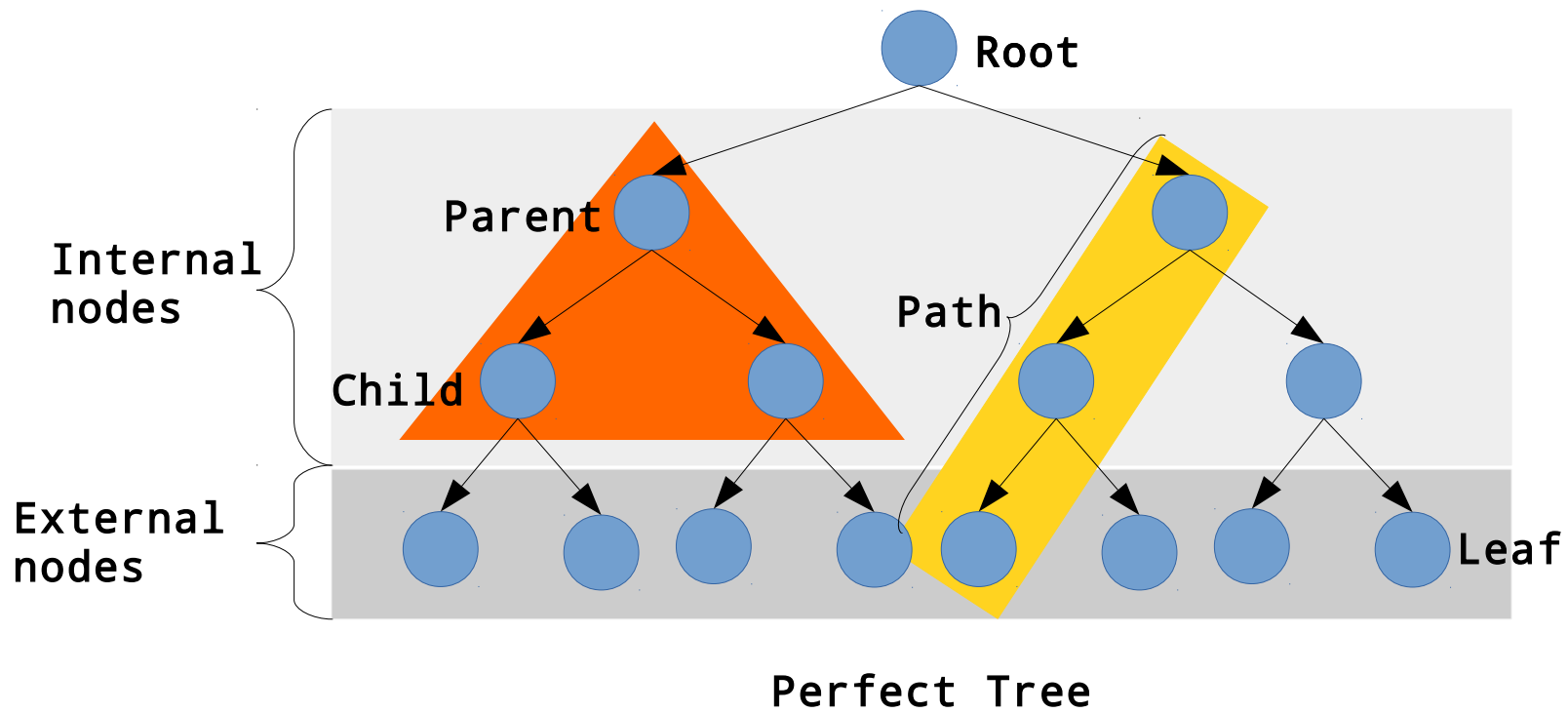


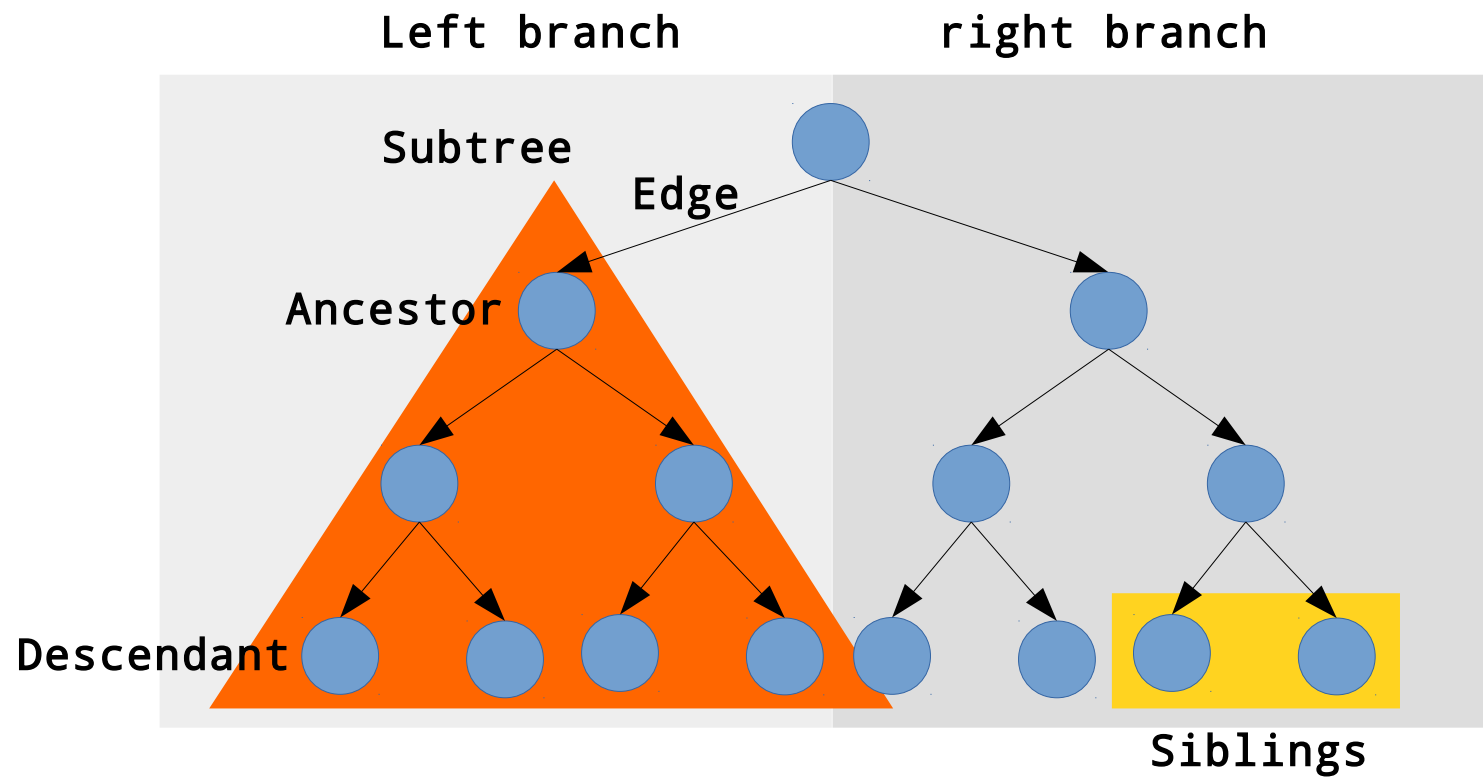
Binary Trees

Angel R. Perez

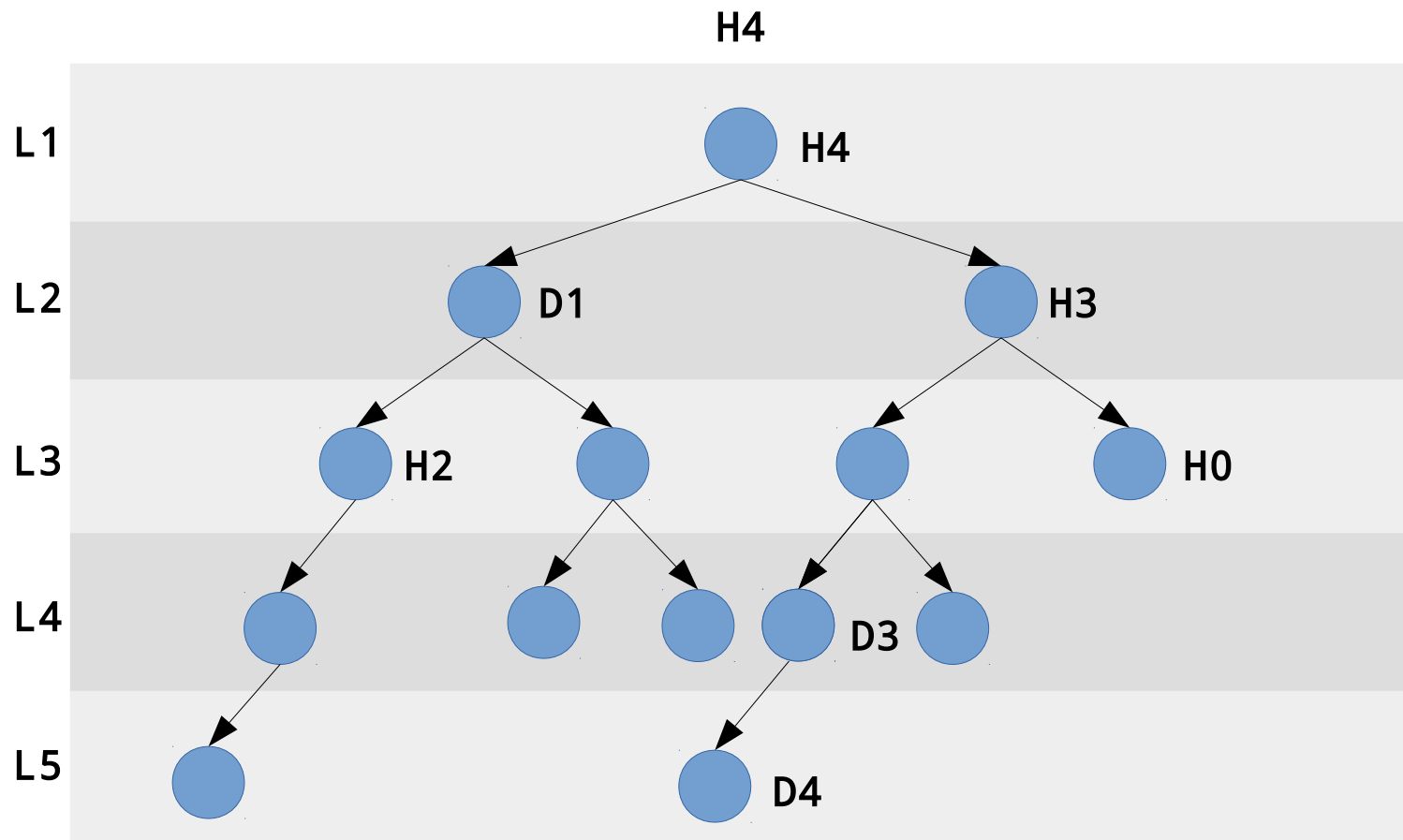
Concepts



Concepts



Concepts



Properties

- No cycles
- Each node has only one direct parent
- A parent only has two or less direct children
- Special case of a directed graph

Operations

- Insertion
- Deletion
- Traversals
 - Pre-Order
 - In-Order
 - Post-Order
 - BFS (breadth first search)
 - DFS (depth first search)
- Search Values

Traversals methods

Pre-Order (root-left-right)

6 4 2 1 3 5 9 7 8 10

In-Order (left-root-right)

1 2 3 4 5 6 7 8 9 10

Post-Order (left-right-root)

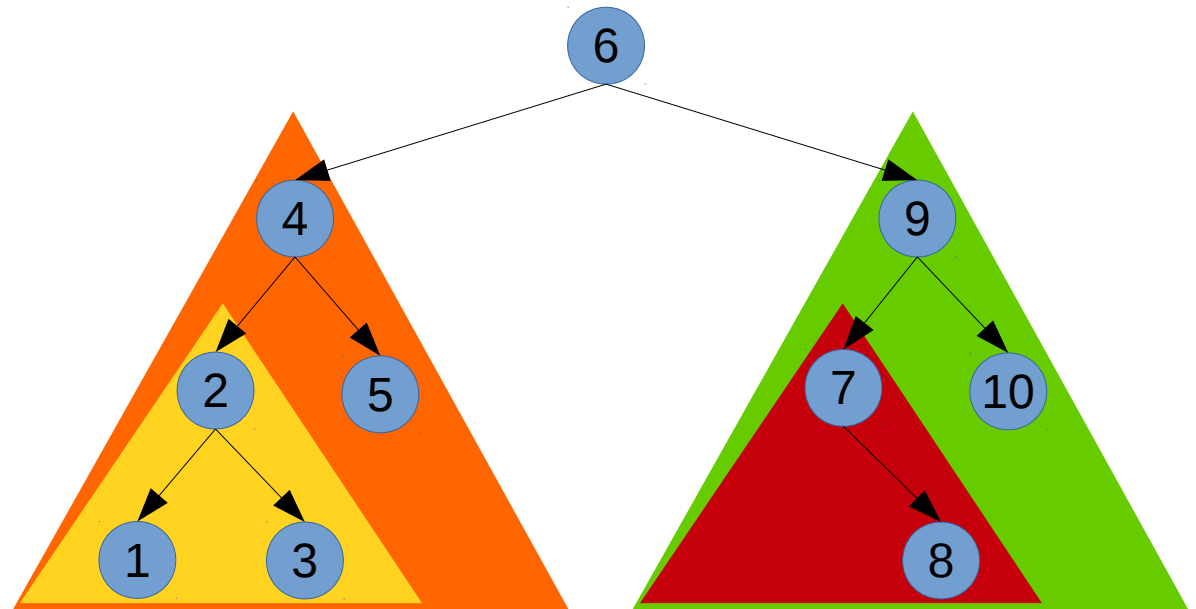
1 3 2 5 4 8 7 10 9 6

BFS (level order)

6 4 9 2 5 7 10 1 3 8

DFS

6 9 10 7 8 4 5 2 3 1



Representation

```
typedef struct node {  
    int data;  
    struct node *left;  
    struct node *right;  
} NODE;
```


Problems

- A binary tree without balancing could degenerate in a linked list.

Challenges

- <https://www.hackerrank.com/challenges/tree-top-view>
 - You need to find a way to print the top view of a tree, this means all the nodes in the left and all the nodes in the right (root included).
- <https://www.hackerrank.com/challenges/binary-search-tree-lowest-common-ancestor>
 - Given two nodes, you need to find what is their common ancestor with lowest value.

SOLUTIONS?

Use Cases

- Schedulers
- Binary search trees
- Sorting
- Encoding

References

- https://en.wikipedia.org/wiki/Binary_tree
- https://en.wikipedia.org/wiki/Breadth-first_search
- https://en.wikipedia.org/wiki/Depth-first_search
- https://en.wikipedia.org/wiki/Tree_%28data_structure%29
- https://en.wikipedia.org/wiki/Tree_traversal
- https://en.wikipedia.org/wiki/Graph_%28abstract_data_type%29