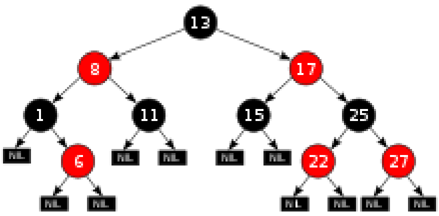
# Branching factor

In computing, tree data structures, and game theory, the **branching factor** is the number of children at each node, the outdegree. If this value is not uniform, an *average branching factor* can be calculated.



A red-black tree with branching factor 2.

For example, in chess, if a "node" is considered to be a legal position, the average branching factor has been said to be about 35.[1][2] This means that, on average, a player has about 35 legal moves at their disposal at each turn. By comparison, the average branching factor for the game Go is 250.[1]

Higher branching factors make algorithms that follow every branch at every node, such as exhaustive brute force searches, computationally more expensive due to the exponentially increasing number of nodes, leading to combinatorial explosion.

For example, if the branching factor is 10, then there will be 10 nodes one level down from the current position, $10^2$ (or 100) nodes two levels down, $10^3$ (or 1,000) nodes three levels down, and so on. The higher the branching factor, the faster this "explosion" occurs. The branching factor can be cut down by a pruning algorithm.

## See also

- Outdegree
- Hierarchy
- Hierarchical organization

## References

1. Levinovitz, Alan (12 May 2014). "The Mystery of Go, the Ancient Game That Computers Still Can't Win" (https://www.wired.com/2014/05/the-world-of-computer-go/). Wired. Retrieved 2014-06-02. "The rate at which possible positions increase is directly related to a game's “branching factor,” or the average number of moves available on any given turn. Chess's branching factor is 35. Go’s is 250. Games with high branching factors make classic search algorithms like minimax extremely costly."
2. Laramée, François Dominic (6 August 2000). "Chess Programming Part IV: Basic Search" (http://www.gamedev.net/page/resources/_/technical/artificial-intelligence/chess-programming-part-iv-basic-search-r1171). GameDev.net. Retrieved 2007-05-01.