Game Analysis

This is extra material (but, yes, it is part of the course), connected with Depth First Search in Chapter 22.

We consider a two person game, call the Players Paul and Carole. There is a set of positions $V$. For each $v \in V$, $MOVE[v]$ is either Paul or Carole and tells you whose turn it is. There is a designated starting position $s$. We have a directed graph $G$ on $V$. For each $v \in V$ there is an adjacency list $Adj[v]$ of those positions $w$ that can be reached in one move. We assume that $V$ is finite and that $G$ has no cycles. A directed graph with no cycles is called a Directed Acyclic Graph, which is abbreviated DAG. We call $v$ an end position if $Adj[v]$ is empty, that is, a leaf in the DAG. For each end position $v$ there is a value $VALUE[v]$. (We are given these VALUES. Oftentimes they are simply 1 or 0, 1 if Paul wins and $-1$ if Carole wins.)

Here is the game. Start at $s$. Players move (remember that $MOVE[v]$ is part of the data so you know who must move) until reaching a leaf $v$. The game is then over and Carole pays Paul $VALUE[v]$ dollars.

Naturally, Paul wants to maximize his payoff and Carole wants to minimize it. But if the game has a million positions how can they analyze it.

DFS provides the key. We apply $DFS-\text{VISIT}[G, s]$. (Any position not reachable from $s$ is clearly irrelevant to the analysis. So lets assume that all of $V$ is reachable from $s$ and that $G$ has $V$ vertices and $E$ edges. Now $E \geq V - 1$ as every position except $s$ came from somewhere. So the time $\Theta(V + E)$ for DFS is actually $\Theta(E)$.) Everytime a vertex $v$ becomes black we define $VALUE[v]$. For the final positions $v$, $VALUE[v]$ is given in the data. So assume $v$ is not a final position. Note, critically, that when $v$ becomes black all $w \in Adj[v]$ have already become black so that, recursively, their $VALUE[w]$ have already been determined. There are two cases: $MOVE[v]$ is Paul. Paul wants to make the move that will maximize his value. So set

$$VALUE[v] = \max_{w \in Adj[v]} VALUE[w]$$

$MOVE[v]$ is Carole. Carole wants to make the move that will maximize his value. So set

$$VALUE[v] = \min_{w \in Adj[v]} VALUE[w]$$

For the extra time, beyond DFS itself, observe that for each $v$ we examine the $w \in Adj[v]$ and so that is a total of $E$ pairs $v, w$ so the additional time is $\Theta(E)$. This is the same order of time as DFS itself. So the game can be
analyzed in time $\Theta(E)$. This works pretty nicely for analyzing Tic Tac Toe. For chess or go the number of possible positions is so large that this analysis is not possible.