Depth-First Search

1. Put the start node on OPEN.
2. If OPEN is empty, exit with failure; otherwise continue.
3. Remove the topmost node from OPEN and put it on CLOSED. Call this node *n*.
4. If the depth of n is equal to the depth bound, clean up CLOSED and go to step 2; otherwise continue.
5. Expand *n*, generating all of its successors. Put these successors (in no particular order) on top of OPEN and provide for each a pointer back to *n*.
6. If any of these successors is a goal node, exit with the solution obtained by tracing back through its pointers; otherwise continue.
7. If any of these successors is a dead end, remove it from OPEN and clean up CLOSED.
8. Go to step 2.

Backtracking. Backtracking is a version of depth-first search that applies the last-in-first-out policy to node *generation* instead of node *expansion*. When a node is first selected for exploration, only one of its successors is generated and this newly generated node, unless it is found to be terminal or dead end, is again submitted for exploration. If, however, the generated node meets some stopping criterion, the program backtracks to the closest unexpanded ancestor, that is, an ancestor still having ungenerated successors. This policy can be implemented by modifying the depth-first search procedure as follows:

The Backtracking Procedure

1. Put the start node on OPEN.
2. If OPEN is empty, exit with failure, otherwise continue.
3. Examine the topmost node from OPEN and call it *n*.
4. If the depth of *n* is equal to the depth-bound or if all the branches emanating from *n* have already been traversed, remove *n* from OPEN and go to step 2; otherwise continue.
5. Generate a new successor of *n* (along a branch not previously traversed), call it *n'*. Put *n'* on top of OPEN and provide a pointer back to *n*.
6. Mark *n* to indicate that the branch (*n*, *n'*) has been traversed.
7. If *n'* is a goal node, exit with the solution obtained by tracing back through its pointers; otherwise continue.
8. If *n'* is a dead end, remove it from OPEN.
9. Go to step 2.

Pearl, Judea
Heuristics.
Addison-Wesley, 1984