

BACKTRACKING

BACKTRACKING ALGORITHMS



Backtracking

- ▶ **IT IS A FORM OF RECURSION !!!**
- ▶ General algorithm for finding all solutions to some computational problems → "constraint satisfaction problems"
- ▶ We incrementally build candidates to the solutions
- ▶ If partial candidate **A** cannot be completed to a valid solution: we abandon **A** as a solution
- ▶ For example: eight-queens problem or sudoku
- ▶ Backtracking is often much faster than brute force enumeration of all complete candidates, because it can eliminate a large number of candidates with a single test
- ▶ Backtracking is an important tool for solving constraint satisfaction problems → combinatorial optimization problems !!!

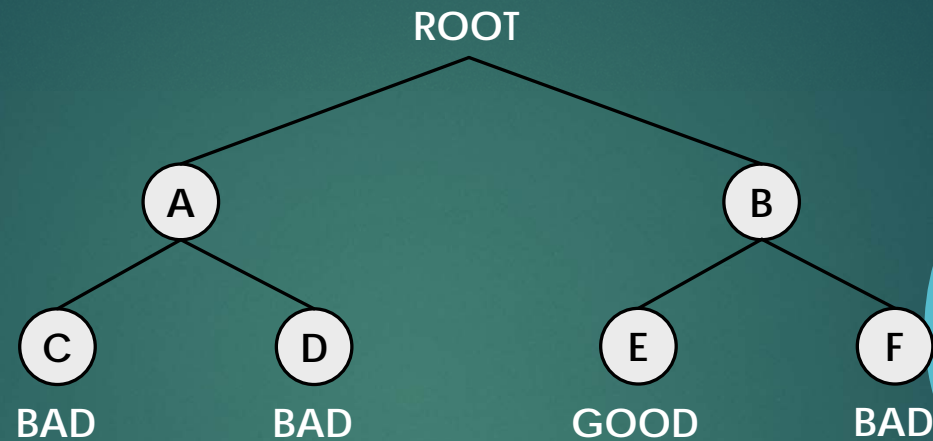
The method

- ▶ The partial candidates are represented as the nodes of a tree structure
- ▶ „*potential search tree*”
- ▶ Each partial candidate is the parent of the candidates that differ from it by a single extension step
- ▶ The leaves of the tree are the partial candidates that cannot be extended any further
- ▶ The backtracking algorithm traverses this search tree recursively, from the root down (like **DFS**)



The method

- ▶ This is why backtracking is sometimes called depth-first search !!!
 - 1.) For every node the algorithm checks whether the given node can be completed to a valid solution
 - 2.) If it can not → the whole subtree is skipped !!!
 - 3.) Recursively enumerates all subtree of the node



- we have several options: we can choose **A** or **B** at the beginning
- after every choice → we have a new set of options
- if we make good choices → we end up with a **GOOD** state
- if not: we have to *backtrack* !!!

