Design and Analysis of Algorithms

Lecture-37

Dharmendra Kumar (Associate Professor)
Department of Computer Science and Engineering
United College of Engineering and Research,
Prayagraj

Backtracking

Backtracking

- **Backtracking** is a general algorithm for finding all (or some) solutions to some computational problems, notably constraint satisfaction problems, that incrementally builds candidates to the solutions, and abandons a candidate ("backtracks") as soon as it determines that the candidate cannot possibly be completed to a valid solution.
- Many problems which deal with searching for a set of solutions or which ask for an optimal solution satisfying some constraints can be solved using the backtracking formulation.

Backtracking

- In the backtrack method, the desired solution is expressible as an n-tuple (x_1, x_2, … , x_n), where the x_i are chosen from some finite set S_i.
- Using backtracking, we find one vector that maximizes(or minimizes or satisfies) a criterion function P (x_1, x_2, … , x_n).
- Sometimes, this method finds all the vectors that satisfies P.
- If m_i is the size of set S_i, then m = m_1m_2…….m_n tuples are possible that satisfy the criterion function P.
- The brute force approach would be to form all these n-tuples, evaluate each one with P, and save those which yield the optimum.
- The backtrack algorithm has as its virtue the ability to yield the same answer with far fewer than m trials.

Backtracking

- Its basic idea is to build up the solution vector one component at a time and to use modified criterion functions P $(x_1, x_2, \ldots, x_i)$. (sometimes called bounding functions) to test whether the vector being formed has any chance of success.

- If the partial vector $(x_1, x_2, \ldots, x_i)$ can in no way lead to an optimal solution, then m_{i+1} ……. m_n possible test vectors can be ignored entirely.

Backtracking

Explicit constraints

Explicit constraints are rules that restrict each x_i to take on values only from a given set.

Common example of explicit constraints are

(1) $x_i \geq 0$ or S_i = The set of all positive real numbers

(2) $x_i = 0$ or 1 or $S_i = \{0, 1\}$

All tuples that satisfy the explicit constraints define a possible solution space.

Implicit constraints

The implicit constraints are rules that determine which of the tuples in the solution space satisfy the criterion function.

Backtracking

State space tree

When we search the solutions of the problem using backtracking, a tree is formed by the result of search. This tree is said to be sate space tree. Each node in the tree represents a state.

Problem state

Each node in the tree is called problem state.

Solution state

Solution states are those problem states s for which the path from root to s defines a tuple in the solution space.

Answer state

Answer states are those solution states s for which the path from the root to s defines a tuple that is a member of the set of solutions of the problem.

Backtracking

State space

All the paths from the root to other nodes define the state space of the problem.

Live node

A node which has been generated and all of whose children have not yet been generated is called a live node.

E-node

The live node whose children are currently being generated is called the E-node (node being expanded).

Dead node

A dead node is a generated node which is not to be expanded further or all of whose children have been generated.

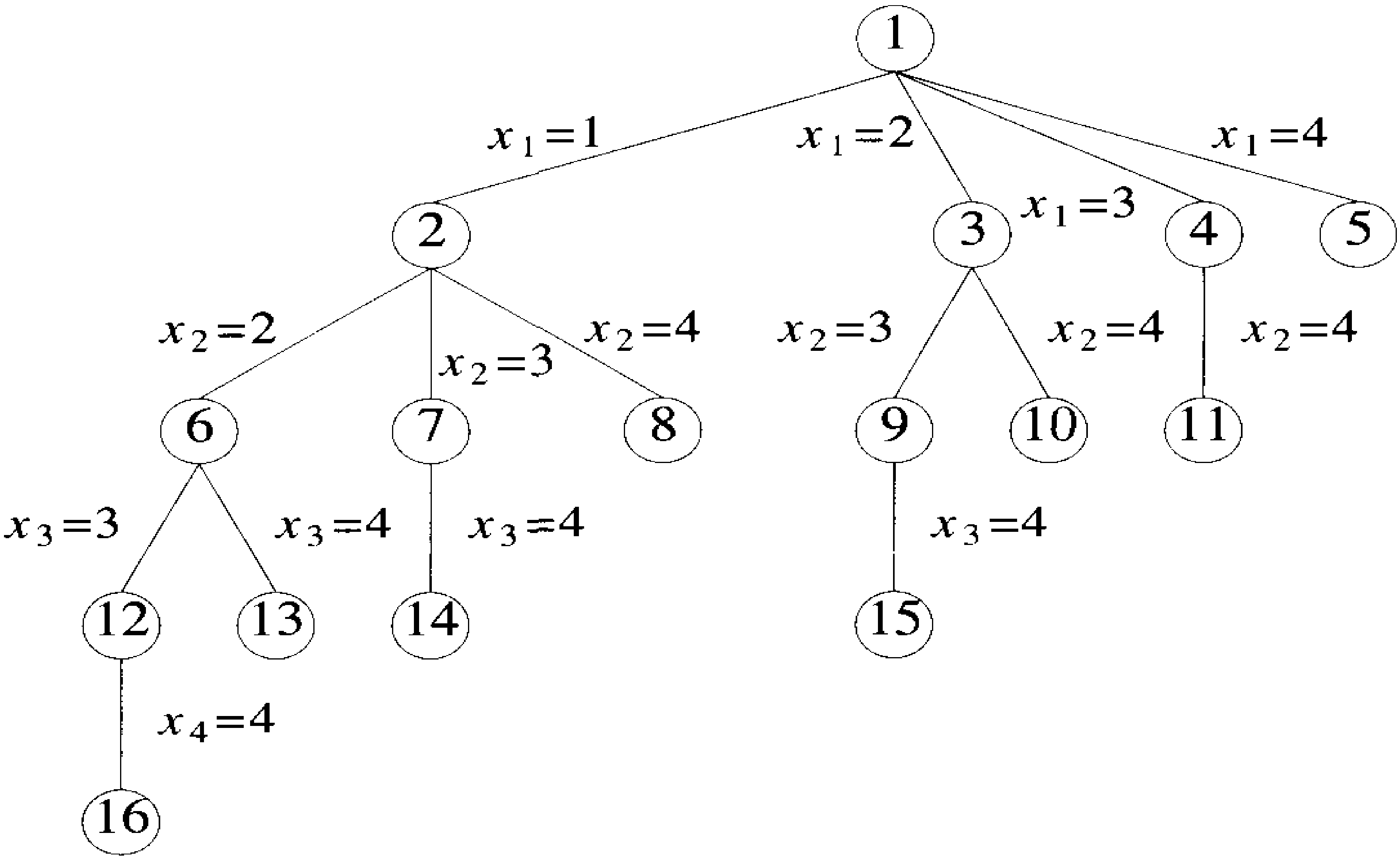
Backtracking

Note: Bounding functions are used to kill live nodes without generating all their children.

Note: Depth first node generation with bounding functions is called backtracking.

Backtracking

Example:



Backtracking

Example: