COT 6405
ANALYSIS OF ALGORITHMS

Backtracking

Computer & Electrical Engineering and Computer Science Department
Florida Atlantic University

Backtracking and Branch-and-Bound

- Both considered improvements over exhaustive search
- Construct candidate solutions one component at a time and evaluate the partially constructed solutions
- If no potential values of the remaining components can lead to a solution, the remaining components are not generated at all
- Based on the construction of a **search tree**; nodes reflect specific choices made for solution's components
- Terminate a node when no solution to the problem can be obtained using node's descendants

Backtracking versus Branch-and-Bound

- Different types of problems:
  - *Branch-and-bound* applicable to optimization problems; based on computing a bound on possible values of the problem's objective function
  - *Backtracking* more often applied to non-optimization problems
- Order in which nodes of the search tree are generated
  - *Backtracking*: tree usually developed depth first (like DFS)
  - *Branch-and-bound* generates nodes using several rules; the most natural one is the *best-first* rule

Backtracking

Search tree:

- The root – initial state before the search for a solutions begins
- Nodes on the first level – choices made for the first component of a solution
- Nodes on the second level – choices for the second component
- So on …
- A node is **promising** if it corresponds to a partially constructed solution that may still lead to a full solution; otherwise it is **nonpromising**
- Leaves are either nonpromising dead ends or complete solutions
- Constructed in a manner of depth-first-search

Backtracking

Search tree construction:

- If the current node is **promising**, its child is generated by adding the first remaining legitimate option for the next component of a solution. Then the process moves to this child.
- If the current node is **nonpromising**, then the algorithm backtracks to the node's parent to consider the next possible option for its last component. If no such option, then it backtracks one more level up in the tree, and so on.
- If a complete solution is found, then the algorithm either stops (if only one solution is required) or it continues searching for other possible solutions.

Problems

- *n*-queens problem
- The Hamiltonian Cycle (HC) problem

Reference: *Algorithms,* R. Johnsonbaugh and M. Schaefer, Pearson Education 2004, chapter 5.4