Ch. 12 – Coping with limitations of algorithm power

* Tackling difficult combinatorial problems (NP-hard problems)
  + Strategy that guarantee to find a solution but may not be in polynomial time.
  + Strategy that find a sub-optimal solution in polynomial time.
* Exact solution strategies
  + Exhaustive search
  + Dp: dynamic programming (everyone hates it, so fuckin hard)
  + Backtracking (wut the hell is dat? Never heard of it)
    - Eliminates some unnecessary cases from consideration
    - Yields solutions in reasonable time for many instances but worst case is still exponential
  + Branch-and-bound
    - Further refines the backtracking idea for optimisation problems

12.1 Backtracking

* Construct the state-space tree
  + Nodes: partial solutions
  + Edges: choices in extending partial solutions
* Explore the state space tree using depth-first search
* “prune” non-promising nodes

12.2 Branch-and-Bound

* An enhancement of backtracking
* Applicable to optimisation problems
* For each node of a state-space tree, compute a bound on the value of the objective function for all descendants of the node
* Uses the bound for
  + Ruling out certain nodes as “non-promising” to prune the tree – if a node’s bound is not better than the best solution seen so far
  + Guiding the search through state-space