Test Data Generation

Given a function and a location we want to reach, how do we derive inputs to the function that lead the control flow to the desired statement?

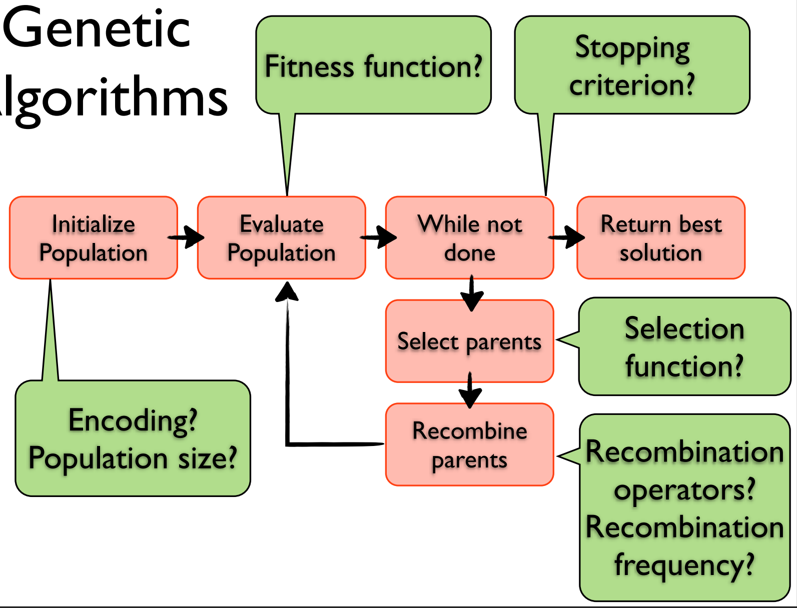
In the examples in previous lectures, we had to think hard to come up with input data that would kill mutants or achieve a coverage goal. For bigger programs, this becomes a tedious thing to do, so we would like to automate it. One approach to do so is to just throw random inputs at the program until we are lucky. Obviously, a better but more costly solution is to search or to analyze the program and deduce the necessary test inputs.

Search-based Software Engineering

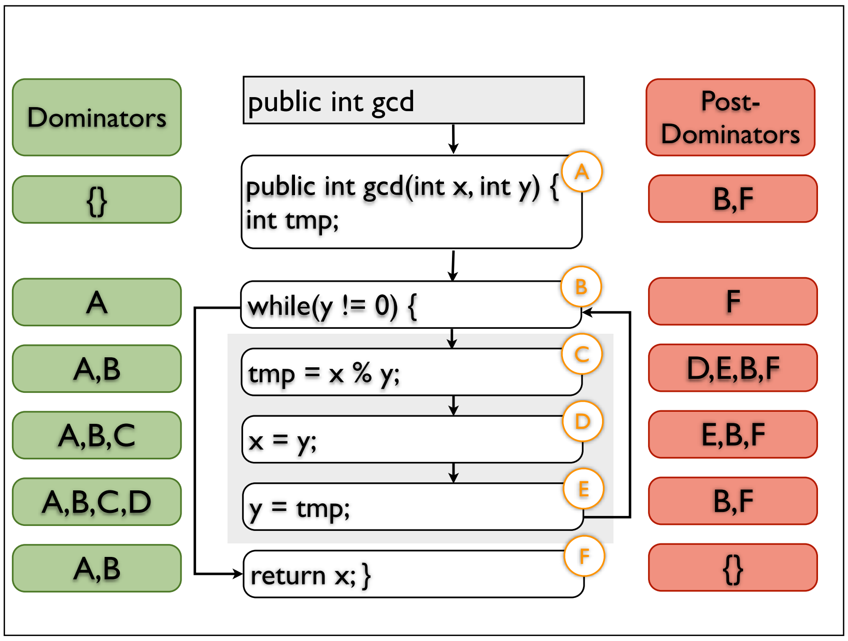
- Cast problems of software engineering as optimization problems
- Search problems in software engineering are BIG
- Apply meta-heuristic search algorithms to solve these problems

Search-based Software Engineering (SBSE) tries to solve software engineering problems by casting them as search problems. As the search space of a typical software engineering problem is huge (remember the roots example?) we can't search exhaustively with e.g. BFS or DFS - we need heuristics.

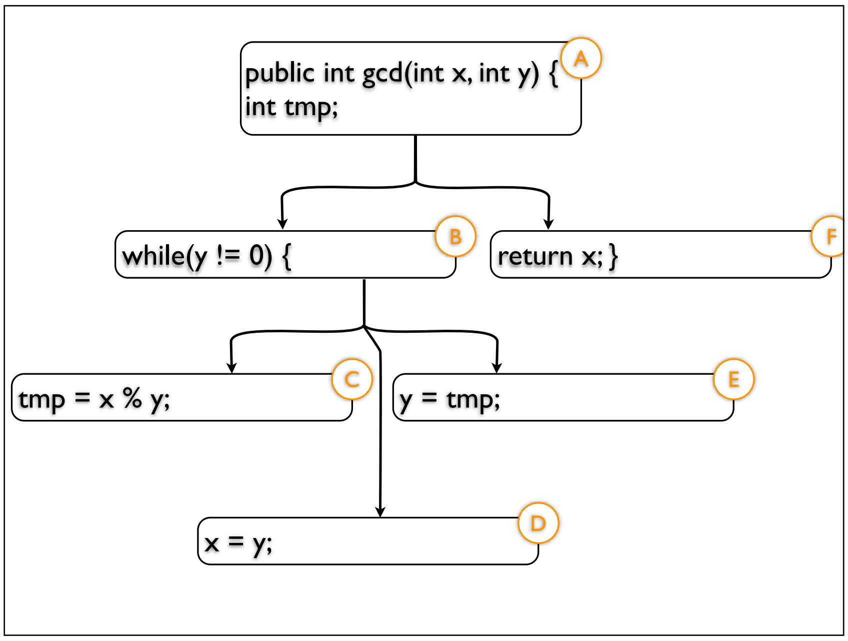
Genetic Algorithms



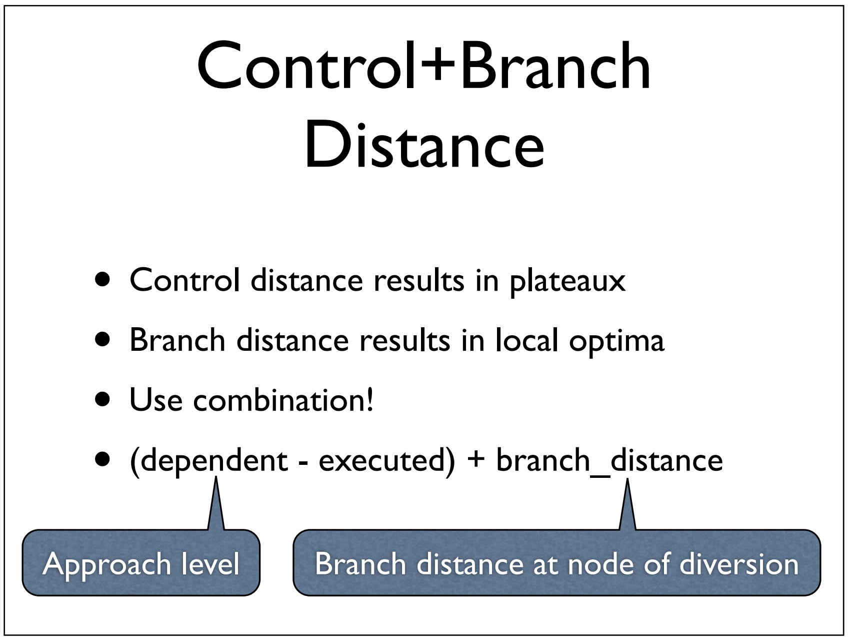
A genetic algorithm is an evolutionary algorithm that evolves a population of candidate solutions towards an optimal solution with respect to some fitness measurement. The algorithm imitates natural selection processes such as crossover and mutation.



A is control dependent on B if: B has at least two successors in the CFG, B dominates A, B is not post-dominated by A, and there is a successor of B that is post-dominated by A



Control-dependence can be represented in a tree: A connection between two nodes in the tree means the child node is control dependent on the parent node.



Control+Branch Distance

- Control distance results in plateaux
- Branch distance results in local optima
- Use combination!
- $\underbrace{\text{(dependent - executed)}}_{\text{Approach level}} + \underbrace{\text{branch\_distance}}_{\text{Branch distance at node of diversion}}$

To cast test data generation as a search problem by looking at the approach level (how many control dependent edges in the CDG are missing?) and the branch distance (how close was the last critical branch to be taken in the right direction?)