* If reject (problemSet, candidateSoln)
  + Return //Go back and try next move
* If accept(problemSet, candidateSoln, numMoves)
  + Return numMoves
* Else
  + S = first(problemSet, candidateSoln)
  + While not empty //(a valid move existed)
    - Backtrack(s)
    - S = next(P, c)