Data Structures II : Backtracking

Mauricio Toro
Department of Informatics and Systems
Universidad EAFIT

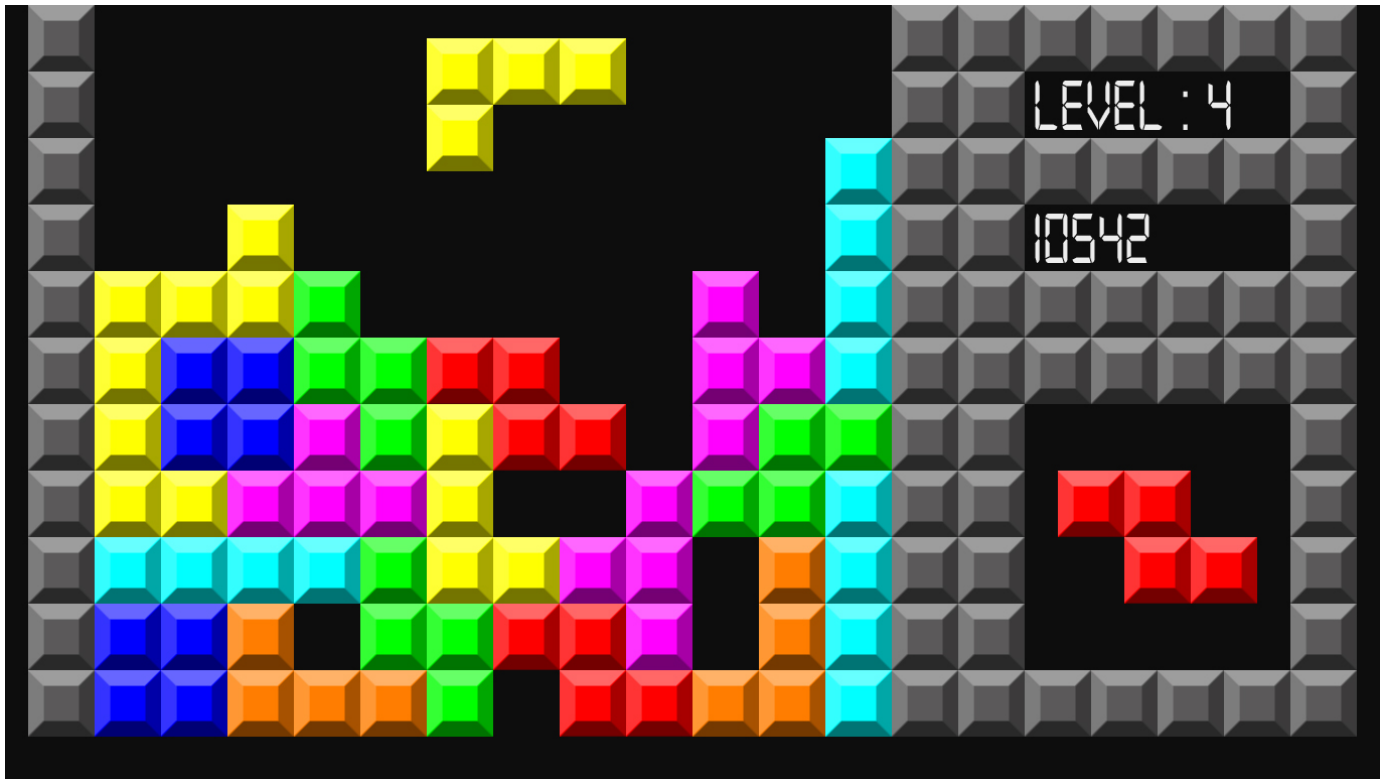
Cocktail of the day: Tequila Sunrise

Disclaimer: Keep alcohol out of the hands of minors.

Cocktail of the day: Tequila Sunrise

- 45 ml of Tequila
- 90 ml of orange juice
- 15 ml of grenadine syrup

Tetris can be solved with Backtracking



http://www.cs.jhu.edu/~susan/600.363/tetris.pdf

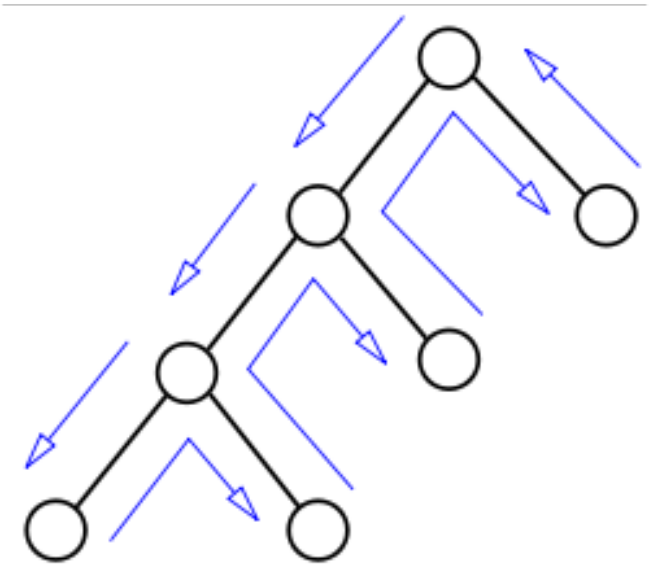
Binary tree

A binary tree is a tree data structure in which each node has at most two children, which are referred to as the left child and the right child.

1. Applications
2. Implementation
3. Some Algorithms
   - Number of elements
   - Maximum Height
   - Search
   - Recursive print

Backtracking

- Backtracking is a general algorithm for finding all (or some) solutions to some computational problems
- Backtracking depends on user-given "black box procedures" that define the problem to be solved.



Applications

- Solving puzzles such as eight queens puzzle, crosswords, Sudoku and Peg Solitaire.
- Combinatorial optimization problems such as the knapsack problem.
  - Constraint satisfaction problems
    My field of research [ADCT11, TAAR09, ORS+11]
- Logic programming languages such as Prolog, which use backtracking internally to generate answers.

Backtracking Algorithm

```
procedure backtracking(c)
  if reject(P,c) then return
  if accept(P,c) then output(P,c)
  s <- first(P,c)
  while s != null do
    backtracking(s)
    s <- next(P,s)
```

Taken from Wikipedia

N Queens

- The *N* queens puzzle asks for all arrangements of eight chess queens on a standard chessboard so that no queen attacks any other.
- Any partial solution that contains two mutually attacking queens can be abandoned, since it cannot possibly be completed to a valid solution.
- https://www.youtube.com/watch?v=Gl75_u4LZU8

N Queens

- The *N* queens puzzle asks for all arrangements of eight chess queens on a standard chessboard so that no queen attacks any other.
- Any partial solution that contains two mutually attacking queens can be abandoned, since it cannot possibly be completed to a valid solution.
- `https://www.youtube.com/watch?v=Gl75_u4LZU8`

N Queens

```
public void placeNqueens(int r, int n) {
    for (int c = 0; c < n; c++) {
        if (canPlaceQueen(r, c)) {
            x[r] = c;
            if (r == n - 1)
                printQueens(x);
            else
                placeNqueens(r + 1, n);
}}}
```

Taken from http://www.java.achchuthan.org/2012/02/n-queens-problem-in-java.html

N Queens: printQueens

```
public void printQueens(int[] x) {
    int N = x.length;
    for (int i = 0; i < N; i++) {
      for (int j = 0; j < N; j++) {
        if (x[i] == j)
            System.out.print("Q ");
        else
            System.out.print("* ");
      }
        System.out.println();
    }
    System.out.println();
}
```

N Queens: canPlaceQueen

```
public boolean canPlaceQueen(int r, int c) {
  for (int i = 0; i < r; i++) {
    if (x[i] == c || (i - r) == (x[i] - c)
                  ||(i - r) == (c - x[i]))
        return false;
  }
        return true;
}
```

References

- Please learn how to reference images, trademarks, videos and fragments of code.
- Avoid plagiarism

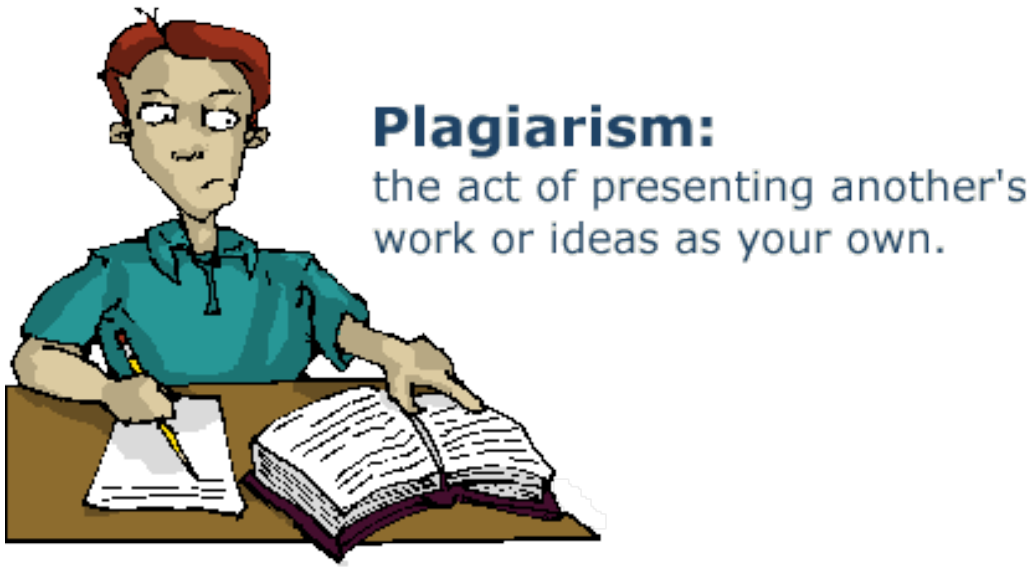


Figure: Figure about plagiarism, University of Malta [Uni09]

References

Antoine Allombert, Myriam Desainte-Catherine, and Mauricio Toro.

Modeling temporal constrains for a system of interactive score.

In Gérard Assayag and Charlotte Truchet, editors, *Constraint Programming in Music*, chapter 1, pages 1–23. Wiley, 2011.

Carlos Olarte, Camilo Rueda, Gerardo Sarria, Mauricio Toro, and Frank Valencia.

Concurrent Constraints Models of Music Interaction.

In Gérard Assayag and Charlotte Truchet, editors, *Constraint Programming in Music*, chapter 6, pages 133–153. Wiley, Hoboken, NJ, USA., 2011.

References II

Mauricio Toro, Carlos Agón, Gérard Assayag, and Camilo Rueda.

Ntccrt: A concurrent constraint framework for real-time interaction.

In *Proc. of ICMC '09*, Montreal, Canada, 2009.

University of Malta.

Plagarism — The act of presenting another's work or ideas as your own, 2009.

[Online; accessed 29-November-2013].