Backtracking

Problemstatement → solution

is a type of tuple

n-tuple $(x_1, \ldots, x_n)$

Criterion Function (Bounding Function)

Constraints

→ Explicit : Talking about what values x_i can take i.e. maths expr

→ Implicit : Describes the way in which x_i must relate to each other..

Where do you save on time compare to brute force?

The major advantage of this method (backtracking):

If it is realized that the partial vector $(x_1, x_1, \ldots, x_i)$ can no way lead to an optimal solution, then $m_{i+1} \ldots m_n$ possible test vectors can be ignored entirely.

All combinations checked

8-Queen Problem

$n=4$

2
4
1
3

All solutions to 8-queens problem represented by 8-tuples $(x_1, \ldots, x_8)$ where x_i is the column on which queen i is placed.

<u>Explicit Constraint:</u>

$1 \le i \le 8$ (x_i itself)

<u>Implicit Constraint:</u>

- No two x_i's can be the same. (x_i to x_{i+1} to relation or x_{i-1} each other)
  (No two queens are in same column)

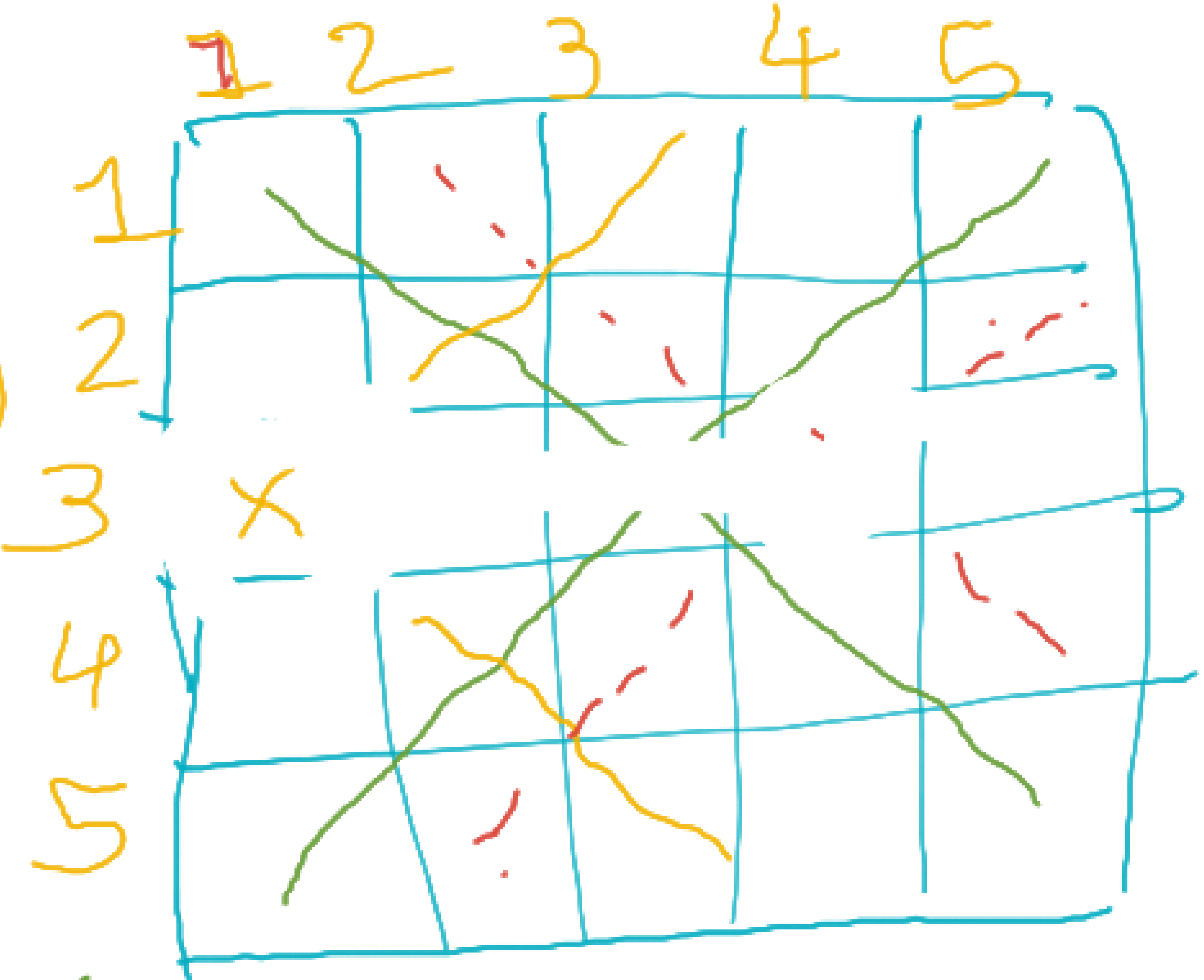
Diagonal Attack

+ve

Primary diagonal
(Left Top to right bottom)
(1,1), (2,2), (3,3), (4,4), (5,5)

Counter diagonal
(1,5) (2,4) (3,3) (4,2) (5,1)
(Right Top to left Bottom)

-ve



4 → (3,1) → (2,2) (1,3) [4, 4]
2 → (4,2) (5,3) [2, 2]

7 → (3,4) → (1,2), (2,3), (4,5) [-1, -1, -1, -1]
→ (2,5), (4,3), (5,2) [7, 7, 7]

+ve slope subtractions should be equal.

-ve slope summation should be equal.

```
Algorithm NQueens(k, n)        total n Queens
{
    for i := 1 to n do
    {
        if Place(k, i) then
        {
            x[k] = i ; //column
                                   queen no/row no
            if (k==n) then write(x[1:n])
            else
                NQueens(k+1, n);          exit?
        }
    }
}
```

Algorithm place(k, i) // How to call

// Returns true if a queen can be placed in k^{th} row and i^{th} column. Otherwise it returns false.

$x[\,]$ is a global array whose first k-1 values have been set.

```
{
  for j=1 to k-1 do
    if ((x[j]=i)   // Two in same column
        OR
        (Abs(x[j]-i) = Abs(j-k)))
    then return false;
} return true;
```

diagonal check

Abs(x) returns the absolute value of x

Problem state : N. Aes

Solution state Feasible

variable
nonfixed

Fixed

answer state

answer state : leaf will be answer state

state space

optimal where the implicit constraints are satisfied.

solution space

state space tree:

the tree organization of the solution space is referred to as the state space tree.