GeeksforGeeks

A computer science portal for geeks

GeeksQuiz

Login

Home Algorithms DS GATE Interview Corner Q&A C C++ Java Books Contribute Ask a Q About

Array Bit Magic C/C++ Articles GFacts Linked List MCQ Misc Output String Tree Graph

Backtracking | Set 1 (The Knight's tour problem)

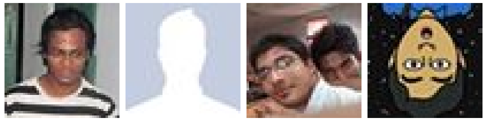
Backtracking is an algorithmic paradigm that tries different solutions until finds a solution that "works". Problems which are typically solved using backtracking technique have following property in common. These problems can only be solved by trying every possible configuration and each configuration is tried only once. A Naive solution for these problems is to try all configurations and output a configuration that follows given problem constraints. Backtracking works in incremental way and is an optimization over the Naive solution where all possible configurations are generated and tried.

For example, consider the following Knight's Tour problem.
The knight is placed on the first block of an empty board and, moving according to the rules of chess, must visit each square exactly once.

Let us first discuss the Naive algorithm for this problem and then the Backtracking algorithm.

Naive Algorithm for Knight's tour
The Naive Algorithm is to generate all tours one by one and check if the generated tour satisfies the constraints.

```
while there are untried tours
{
   generate the next tour
   if this tour covers all squares
   {
      print this path;
   }
}
```



Interview Experiences
Advanced Data Structures
Dynamic Programming
Greedy Algorithms
Backtracking
Pattern Searching
Divide & Conquer
Mathematical Algorithms
Recursion
Geometric Algorithms

Backtracking works in an incremental way to attack problems. Typically, we start from an empty solution vector and one by one add items (Meaning of item varies from problem to problem. In context of Knight's tour problem, an item is a Knight's move). When we add an item, we check if adding the current item violates the problem constraint, if it does then we remove the item and try other alternatives. If none of the alternatives work out then we go to previous stage and remove the item added in the previous stage. If we reach the initial stage back then we say that no solution exists. If adding an item doesn't violate constraints then we recursively add items one by one. If the solution vector becomes complete then we print the solution.

Backtracking Algorithm for Knight's tour

Following is the Backtracking algorithm for Knight's tour problem.

```
If all squares are visited
    print the solution
Else
   a) Add one of the next moves to solution vector and recursively
   check if this move leads to a solution. (A Knight can make maximum
   eight moves. We choose one of the 8 moves in this step).
   b) If the move chosen in the above step doesn't lead to a solution
   then remove this move from the solution vector and try other
   alternative moves.
   c) If none of the alternatives work then return false (Returning false
   will remove the previously added item in recursion and if false is
   returned by the initial call of recursion then "no solution exists" )
```

Following is C implementation for Knight's tour problem. It prints one of the possible solutions in 2D matrix form. Basically, the output is a 2D 8*8 matrix with numbers from 0 to 63 and these numbers show steps made by Knight.

```
#include<stdio.h>
#define N 8

int solveKTUtil(int x, int y, int movei, int sol[N][N], int xMove[],
                int yMove[]);

/* A utility function to check if i,j are valid indexes for N*N chessb
int isSafe(int x, int y, int sol[N][N])
{
```

Popular Posts

All permutations of a given string
Memory Layout of C Programs
Understanding "extern" keyword in C
Median of two sorted arrays
Tree traversal without recursion and without stack!
Structure Member Alignment, Padding and Data Packing
Intersection point of two Linked Lists
Lowest Common Ancestor in a BST.
Check if a binary tree is BST or not
Sorted Linked List to Balanced BST

```
    if ( x >= 0 && x < N && y >= 0 && y < N && sol[x][y] == -1)
        return 1;
    return 0;
}

/* A utility function to print solution matrix sol[N][N] */
void printSolution(int sol[N][N])
{
    for (int x = 0; x < N; x++)
    {
        for (int y = 0; y < N; y++)
            printf(" %2d ", sol[x][y]);
        printf("\n");
    }
}

/* This function solves the Knight Tour problem using Backtracking.  T
function mainly uses solveKTUtil() to solve the problem. It returns fa
no complete tour is possible, otherwise return true and prints the tou
Please note that there may be more than one solutions, this function
prints one of the feasible solutions.  */
bool solveKT()
{
    int sol[N][N];

    /* Initialization of solution matrix */
    for (int x = 0; x < N; x++)
        for (int y = 0; y < N; y++)
            sol[x][y] = -1;

    /* xMove[] and yMove[] define next move of Knight.
       xMove[] is for next value of x coordinate
       yMove[] is for next value of y coordinate */
    int xMove[8] = {  2, 1, -1, -2, -2, -1,  1,  2 };
    int yMove[8] = {  1, 2,  2,  1, -1, -2, -2, -1 };

    // Since the Knight is initially at the first block
    sol[0][0]  = 0;

    /* Start from 0,0 and explore all tours using solveKTUtil() */
    if(solveKTUtil(0, 0, 1, sol, xMove, yMove) == false)
    {
        printf("Solution does not exist");
        return false;
    }
    else
        printSolution(sol);
    return true;
}

/* A recursive utility function to solve Knight Tour problem */
int solveKTUtil(int x, int y, int movei, int sol[N][N], int xMove[N],
                int yMove[N])
{
   int k, next_x, next_y;
   if (movei == N*N)
       return true;

   /* Try all next moves from the current coordinate x, y */
   for (k = 0; k < 8; k++)
   {
       next_x = x + xMove[k];
       next_y = y + yMove[k];
       if (isSafe(next_x, next_y, sol))
       {
         sol[next_x][next_y] = movei;
         if (solveKTUtil(next_x, next_y, movei+1, sol, xMove, yMove) =
             return true;
         else
             sol[next_x][next_y] = -1;// backtracking
       }
   }

   return false;
}

/* Driver program to test above functions */
int main()
{
    solveKT();
    getchar();
    return 0;
}
```

Output:

```
 0  59  38  33  30  17   8  63
37  34  31  60   9  62  29  16
58   1  36  39  32  27  18   7
35  48  41  26  61  10  15  28
42  57   2  49  40  23   6  19
```

Recent Comments

Abhi You live US or India?

Google (Mountain View) interview · 8 minutes ago

Aman Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 47 minutes ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 51 minutes ago

Sanjay Agarwal bool tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 1 hour ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 1 hour ago

newCoder3006 If the array contains negative numbers also. We...

Find subarray with given sum · 1 hour ago

```
 47  50  45  54  25  20  11  14
 56  43  52   3  22  13  24   5
 51  46  55  44  53   4  21  12
```

Note that Backtracking is not the best solution for the Knight's tour problem. See this for other better solutions. The purpose of this post is to explain Backtracking with an example.

References:

http://see.stanford.edu/materials/icspacs106b/H19-RecBacktrackExamples.pdf
http://www.cis.upenn.edu/~matuszek/cit594-2009/Lectures/35-backtracking.ppt
http://mathworld.wolfram.com/KnightsTour.html
http://en.wikipedia.org/wiki/Knight%27s_tour

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Related Tpoics:

- Backtracking | Set 8 (Solving Cryptarithmetic Puzzles)
- Tail Recursion

- Find if two rectangles overlap
- Analysis of Algorithm | Set 4 (Solving Recurrences)
- Print all possible paths from top left to bottom right of a mXn matrix
- Generate all unique partitions of an integer
- Russian Peasant Multiplication
- Closest Pair of Points | O(nlogn) Implementation

22 Tweet 0 3

Writing code in comment? Please use **ideone.com** and share the link here.

35 Comments GeeksforGeeks

Sort by Newest ▾

Join the discussion…

arjomanD • 3 days ago

Hello Every1 . I had a question . i have my code BUT it works very very Slow f answer !

Would you Plz help me make my code and algorithm better ?

http://paste.ubuntu.com/742019...

Thanks !

KC • 6 months ago

Knight moves 2 squares in the direction of one of the axes of the board, and th This definition can then be generalized to obtain an (a, b) knight. If the number possible in non 8x8 board using this similar algorithm.

Probe · 7 months ago

Here is also my implementation of the above algorithm for java:

```
[code]public class HorseTraversing{

public static void printBoard(int[][] board){
for (int i=0; i<board.length; i++){="" for="" (int="" y="0;" y<board.length;="" y++
System.out.print(board[i][y] + " ");
}else {
System.out.print(" " + board[i][y] + " ");
}
}
System.out.println();
}
}

public static boolean isSafeMove(int moveX, int moveY, int[][] board){
int maxPosition = (board.length - 1);

if (moveX <= maxPosition && moveY <= maxPosition && moveX >= 0
```

see more

dark_night · 9 months ago

I am still not getting the difference between backtracking and recursion.. espec

Here also we are using every possible move and in recursion too we do the sa more efficient than recursion.

1

Let's Make a New India · 9 months ago

Muhammad Barrima thanks s a lot 4 this kind of help 4 geek like me

Let's Make a New India · 9 months ago

Muhammad Barrima thanks s a lot 4 this kind of help 4 geek like me

shashank · 10 months ago

How to decide xMove[8] and yMove[8] arrays contents?

Dev Khanna ➜ shashank · 8 months ago

They are the contents of the possible moves from any given square. F
left one and up two. The corresponding "vector" would be (1, 2)

Mahendra Mundru · 10 months ago

It seems to be O(N power (N**2)).

coder · 11 months ago

can anyone tell me why this code works so slow

```
 #include<iostream>
#define max 8
using namespace std;
void printsol(int arr[][max])
{
    int i,j;
    for(i=0;i<=max-1;i++)
    {
        for(j=0;j<=max-1;j++)
        cout<<arr[i][j]<<' ';
        cout<<'\n';
    }
    return ;
}
bool issafe(int arr[][max],int i,int j)
{
```

see more

Guest coder • 4 months ago

There is a reason why geekforgeeks folks chose
xMove[8] = { 2, 1, -1, -2, -2, -1, 1, 2 }
yMove[8] = { 1, 2, 2, 1, -1, -2, -2, -1 }

Only for this configuration of steps by knight, the recursion runs faster later stages.

For any other configuration like
xMove[8] = { 2, 2, 1, 1, -2, -2, -1, -1 }
yMove[8] = { 1, -1, 2, -2, 1, -1, 2, -2 }

it's much much slower.

But the 2nd configuration is correct too. It's just that it takes more time

1

zaraki • a year ago

Hi,
Maybe I am missing something. I did code this up to get the exact same outpu
I believe 0 represents the leftmost square and 63 the rightmost.

The order that is printed does not seem to be how a knight would move:
row wise: 0(0,0) 59(7,3) 38(4,6) 33 30 17 8 63

row-wise: 0(0,0) 59(7,3) 38(4,6) 33 30 17 8 63

if I take it columnwise still the coordinates seem too far. Please let me know if other order of the tour that I am missing.
Thanks!

Priyanka ➦ zaraki • 11 months ago

Hi,

the values being printed here in the solution tells the order of Knight mo
Square with value 0 means knight will start from this square.
Square with value 1 means knight will move to this square next.
Square with value 2 means knight will move to this square next and so

Since we are moving 64 times (0 to 63) it signifies the solution covers
solution.
Hope this helps.

yashraj • a year ago

```
#include
#define N 8
typedef enum{false,true} bool;

int solveKTutil(int sol[N][N],int nx[N],int ny[N],int x,int y,int nmove);

int is_safe(int sol[N][N],int x,int y)
{
if(x>=0 && x=0 && y<N && sol[x][y]==-1)
return 1;
return 0;

}
```

```
,
void print_sol(int sol[N][N])
{
int i,j;
for(i=0;i<N;i++)
{
for(i=0;i<N;i++)
```

see more

Ajeet Singh Yadav • a year ago

thanks a lot :)

Muhammad Barrima • a year ago

the knight on a chess board moves in this way
2 moves up and then 1 right
2 moves up and then 1 left
2 moves down and then 1 right
2 moves down and then 1 right
2 moves right and then 1 up
2 moves right and then 1 down
2 moves left and then 1 up
2 moves left and then 1 down

Then he made those arrays to check for all possible next moves :)

Ajeet Singh Yadav • a year ago

can somebody please explain the.

int xMove[8] = { 2, 1, -1, -2, -2, -1, 1, 2 };.

int yMove[8] = { 1, 2, 2, 1, -1, -2, -2, -1 };.

please do explain how to write these?

Abhinav Priyadarshi · 2 years ago

one very interesting thing to notice here is that changing the order of moves in
amount.

here (http://ideone.com/gzVrt) is the orignal code executing in 0.48 seconds.
while this(http://ideone.com/qfsf3) one is taking huge amount of time(time lim
moves has been changed.

nitin gupta · 2 years ago

```
 /* can some body tell me why?
int xMove[8] = {  2, 1, -1, -2, -2, -1,  1,  2 };
    int yMove[8] = {  1, 2,  2,  1, -1, -2, -2, -1 };
are like that ...wt it means?*/
```

nitin gupta · 2 years ago

can some body tell me ...XMove and YMove array ke content aise hi kyo hai ?

```
 /* Paste your code here (You may delete these lines if not writing co
```

anant · 2 years ago

@geeksforgeeks:
please tell(or if possible explain), the time complexity for knight tour using back

1

kartik anant • 2 years ago

@anant: Getting a tight upper bound for such problems is never easy. though.

The recursion tree will be a tree of depth 64. Every internal node will ha the upper bound on time complexity is 1 + 8 + 8*8 + 8*8*8 + ... (64 tim

More expert comments from other users are welcome!

Dedicated Programmer • 2 years ago

I have used this code, but unable to figure out why it is printing same solution r

Dedicated Programmer • 2 years ago

```
 /* Paste your code here (You may delete these lines if not writing co
#define N 5

int isValidPosition(int (*Table)[N],int row,int col)
{
        if(row>=0 && row<N && col>=0 && col<N && !Table[row][col])
                return 1;
        return 0;
}

void printSolution(int (*Table)[N],int top)
{
        int i,j;

        for(i=0;i<N;i++)
        {
```

```
        {
                for(j=0;j<N;j++)
                        printf("%d ",Table[i][j]);
```

see more

Ajinkya • 2 years ago

What is the time complexity of this??
and also please comment on the time complexity of a backtracking solution in
problem, but if possible discuss complexities of 8 queens problem, rat maze p
Thanks a ton...
Geeksforgeeks is savior and a great educator! :)

```
/* Paste your code here (You may delete these lines if not writing co
```

vivek • 3 years ago

Additional information about topic:

Knight's tour is a variation of Hamiltonian path problem in graph theory.
Knight's tour is a NP complete problem.

anantha89 • 3 years ago

Hi All,

When I tried with,
int a[8] = {-2, +2, +1, +1, -2, +2, -1, -1};
int b[8] = {+1, +1, -2, +2, -1, -1, -2, +2};

It did not solve.

when I used
int a[8] = {2, 1, -1, -2, -2, -1, 1, 2};
int b[8] = {1, 2, 2, 1, -1, -2, -2, -1};

It solved in less time.

How?

yashraj ➦ anantha89 • a year ago

i think the number of waste moves with the second one are far less co
of backtracks are less in second case.. so it is running fast towards th

```
/* Paste your code here (You may delete these lines if not writ
```

alan • 3 years ago

@geeksforgeeks
In your back-traking code, how are you making sure that 'knight ends on a squ
began'.

Sandeep ➦ alan • 3 years ago

@alan: There was an error in post. The code and algorithm actually pr
be open or close. I have updated the post.

asd • 3 years ago

Is this the best possible solution. ? Is there any better solution ?

Sandeep ➦ asd • 3 years ago

@asd: The above solution is not the best solution for Knight's tour prol solutions. The purpose of this post is to explain Backtracking with an e

I have added same note to the original post.

shanker ➦ Sandeep • 3 years ago

@sandeep can you post the solution of 8 queen,knapsack,parti it up

shanker ➦ Sandeep • 3 years ago

@sandeep can you post the solution of 8 queen,knapsack,parti it up

Sandeep ➦ shanker • 3 years ago

@shankar: Thanks! We will be covering more problems

Subscribe   Add Disqus to your site