

ALFILES

Sara Chica, Rodrigo Gualtero

29 de Octubre, 2012

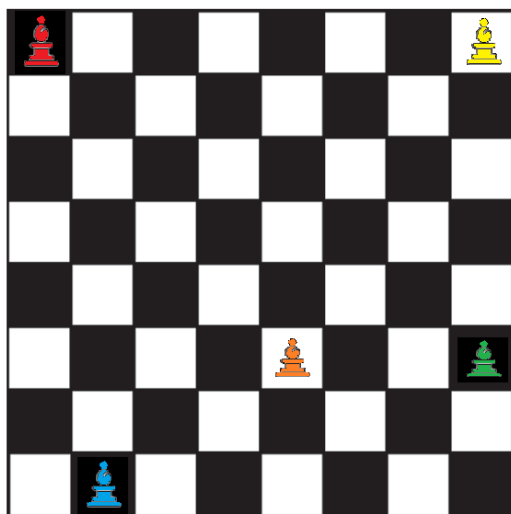
Índice

1. Introducción	1
2. Definición del problema	3
2.1. Entrada	3
2.2. Salida	3
3. Modelamiento matemático	3
4. Planteamiento de la Solución	3
5. Conclusiones	4

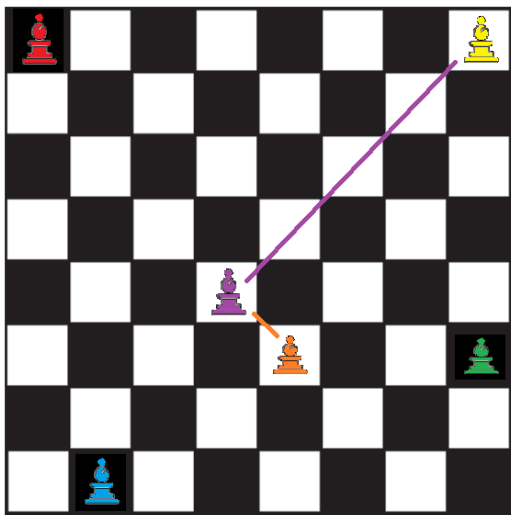
1. Introducción

Este es un problema de la UVA, identificado con el código *861*, El cual consiste en encontrar el número de posibilidades en que se pueden distribuir K alfiles en una tabla de ajedrez de tal manera que no sea posible que un alfil pueda sacar a otro alfil del juego.

A continuación se presenta un ejemplo en donde se puede observar más claramente lo dicho anteriormente:



La imagen representa un tablero de ajedrez en donde se encuentran distribuidos 5 alfiles de tal manera que ningún alfil puede sacar a otro del juego.



Notese que en la imagen tenemos 6 alfiles pero a diferencia del ejemplo anterior los alfiles están organizados de tal forma que un alfil puede sacar a otro alfil del juego.

2. Definición del problema

Para este problema se debe encontrar la cantidad de posibilidades en que se pueden organizar los alfiles en un tablero de ajedrez de tal manera que ningún alfil pueda sacar a otro; este problema se puede modelar como un árbol de posibilidades.

2.1. Entrada

En un principio recibimos dos enteros; el primer entero representado con la letra n ($1 \leq n \leq 8$) que indica la dimensión del tablero de ajedrez y K ($1 \leq K \leq n^2$) representando el número de alfiles que se organizarán en el tablero.

2.2. Salida

Por cada caso de prueba se debe imprimir una línea que contenga el número total de posiciones en que se pueden colocar los alfiles sobre un tablero dado, de tal forma que no haya dos de ellos en situación de posible ataque.

3. Modelamiento matemático

Es importante saber cuando dos alfiles según su posición están en situación de ataque lo cual se puede definir matemáticamente de la siguiente forma:

Sean K_i y K_j alfiles, donde K_i tiene posiciones (X_1, Y_1) y K_j tiene posiciones (X_2, Y_2) .

Se dice que K_i puede atacar a K_j si:

$$(|X_2 - X_1|) = (|Y_2 - Y_1|)$$

4. Planteamiento de la Solución

Para determinar la solución del problema se deben emplear varios algoritmos de recorrido sobre árboles con el fin de encontrar todas las posibilidades de ubicación de los alfiles; estos algoritmos se conocen comunmente bajo el tema de rastreo exhaustivo o rastreo exhaustivo por retroceso.

Lo que se debe saber acerca de los algoritmos es que por su complejidad solo son válidos en espacios de resultados que no superen las $8!$ posibilidades ya que generalmente las máquinas no pueden procesar esto.

Los algoritmos empleados para la solución se describen de forma general a continuación.

Algoritmo Backtracking o vuelta atras: Este algoritmo consiste en realizar búsquedas sistemáticas sobre el árbol de tal forma que separa esta búsqueda en sub tareas para ir generando soluciones parciales al problema a medida que progresa en el recorrido.

Algoritmo Candidates: Este algoritmo va muy de la mano con el algoritmo anterior ya que el funcionamiento del Backtracking depende de este algoritmo; el objetivo del candidate es ir como su nombre lo indica buscando posibles candidatos de solución para el problema. Para realizar este algoritmo adecuadamente es importante tener en cuenta lo dicho en el planteamiento del problema.

5. Conclusiones

1. Es importante investigar sobre este tipo de algoritmos de recorridos sobre árboles porque permiten abarcar todas las posibilidades para solucionar un problema.
2. Estos algoritmos que resuelven problemas en donde se debe encontrar todas las posibilidades de éxito tienen aplicaciones en probabilidad y estadística.
3. Este tipo de problemas permite que los estudiantes se den cuenta de lo importante que puede ser para un sistema informático conocer todos los posibles caminos que puede abordar si lo que se quiere es evitar fallas contemplando todas las posibilidades que pueden generarlas.