|  |
| --- |
| PONTIFICia universidad javeriana cali |
| Mini proyecto Análisis y Diseños de Algoritmos |
| NumberLink |
|  |
| **Diego Lozada** |
|  |

**Camilo Quintero**

**02/06/2014**

|  |
| --- |
| Creación de una aplicación que soluciona el juego Numberlink por los métodos de Backtracking, fuerza bruta y voraz, con el fin de sustenta el mini proyecto del curso de análisis y diseño de algoritmo de la Pontificia Universidad Javeriana Cali |

Contenido

[Análisis de la solución 3](#_Toc389521518)

[Backtracking 3](#_Toc389521519)

[Fuerza bruta 3](#_Toc389521520)

[Voraz 3](#_Toc389521521)

[Detalles principales de implementación 4](#_Toc389521522)

[Backtracking 4](#_Toc389521523)

[Fuerza bruta 4](#_Toc389521524)

[Voraz 4](#_Toc389521525)

# Análisis de la solución

## Backtracking

## Fuerza bruta

Primero se verifico que el tablero tuviera correctamente los nodos (números a conectar), luego se guardan estos nodos es una estructura la cual permite el reconocimiento exacto en el tablero de un nodo inicial y uno terminal (no importa el orden solo para explicar), luego se ejecuto un recorrido casilla por casilla y se cambia el dato con respecto a la lista de nodos del tablero teniendo en cuenta si la casilla visitada es vacía, si lo anterior es correcto se genera una nueva alternativa.

Estas alternativas se guardan en una estructura cola , y siempre que visita una nueva casilla, se repite el proceso tomando como base la primera alternativa en la estructura cola, que luego de modificarse, quedar como una nueva alternativa y pasa a ser ultima en la cola.

Luego de que se recorran toda la tabla, se envía una a una las alternativas (en este momento son soluciones) generas por el método al verificador para contar las soluciones correctas.

Se muestra como alternativa de solución a la última solución verificada.

## Voraz

Primero se verifico que el tablero tuviera correctamente los nodos (números a conectar), luego se guardan estos nodos es una estructura la cual permite el reconocimiento exacto en el tablero de un nodo inicial y uno terminal (no importa el orden solo para explicar).

Luego se crea un tablero de frecuencia, el cual ayudado por el tablero de alternativas a identifica si un camino se ramifico, además ayuda a crear las soluciones basadas en frecuencia de los vecinos (arriba, abajo, derecha, izquierda).

Luego del proceso anterior se procederá a recorrer cada nodo del tablero de alternativas (tanto terminal como inicial), estos tableros están guardados en una estructura cola (empezando por el tablero inicial), y se crearan alternativas teniendo en cuenta el nodo y la frecuencia del mismo con respecto a sus vecinos, cada alternativa modificara la actual y se moverá a la ultima posición de la estructura cola que guarda las alternativas.

Cada alternativa guardara en una nueva estructura los nodos pendientes a visitar, si una alternativa se queda sin nodos a visitar, se recorrerá el tablero alternativa y se reconocerán nuevos nodos teniendo en cuenta la frecuencia del nodo

Cada alternativa se verificara y se contara si es una solución válida.

Se muestra como alternativa de solución a la última solución verificada.

# Detalles principales de implementación

## Backtracking

Para la solución por método del backtracking. La cantidad de soluciones dado un tablero de y donde la cantidad de números pares a conectar es de , es de: 1 para todos las entradas probadas.

No siempre hay solución.

## Fuerza bruta

Para la solución por método de fuerza bruta. La cantidad de soluciones dado un tablero de y donde la cantidad de números pares a conectar es de , es de: para cualquier entrada.

para este método solo genera las soluciones anteriores si el tablero esta por lo menos con sus parejas completas.

## Voraz