Laboratorio Nro. 2
Fuerza bruta (Brute force o Exhaustive search)

Miguel Angel Martinez Florez
Universidad Eafit
Medellín, Colombia
mamartinef@eafit.edu.co

Daniel Ricardo Palacios Diego
Universidad Eafit
Medellín, Colombia
drpalaciod@eafit.edu.co

Pablo Micolta López
Universidad Eafit
Medellín, Colombia
pmicoltal@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1 En nuestro código utilizamos un diccionario para acomodar las id de los lugares dados en el mapa. Además, estas claves se guardaron en una lista con n+1 espacios, tomando como lugar de inicio el primer nodo que nos dan, el cual se puso al inicio y al final de la lista; y con los que quedaron en el medio se sacaron todas las posibles permutaciones, las cuales se comprobaron que estaban en un orden posible, es decir, que todos los puntos de la lista eran sucesores de su anterior y luego de todos los que cumplían este criterio se busco por los que tenían un peso total menor.

3.2 La complejidad asintótica es de $O(V!^2*V*A)$

3.3 Como tal es imposible saber el tiempo total con solo los vértices pues necesitamos saber las aristas que hay también pero asumiendo el caso con menor cantidad de aristas (50 aristas pues se necesitan 1 para conectar todos los puntos 1 vez y otra para conectar el nodo de inicio con el ultimo nodo) el total de operaciones hacer son de $2.3*10^{132}$ lo que hace de esta solución no aplicable pues asumiendo un mínimo promedio de 100000000 operaciones por segundo este programa se mas de 200 000 septillones de segundos

3.4 nosotros guardamos el tablero como nos lo dibujaban, es decir, los puntos(.) y los asteriscos (*), en una lista de strings, donde cada string representa una línea, mientras tanto la posición en la que se va a poner la reina se representa con un arreglo que tiene números del 0 hasta el tamaño que nos están pidiendo -1. En este arreglo, el índice representa la columna en la que esta la reina y el numero guardado en esa posición es la fila en donde

PhD. Mauricio Toro Bermúdez
Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473

está la reina. Esto nos permite un fácil acceso a ambos valores lo cual es importante para comprobar la validez de una solución, además utilizamos una lista para guardar las coordenadas en las que se encuentran los puntos inutilizables, ya que, no sabemos cuántos de estos puede haber. Con esto lo que hicimos fue generar todas las permutaciones posibles de las posiciones y posteriormente pasamos a validar cada una de estas permutaciones a través de dos operaciones; la primera revisa que las pendientes de todos los puntos del arreglo no sean iguales a 1, ya que, si lo son se atacaran a las reinas, además se comprueba que no halla dos reinas en la misma fila; y el segundo revisa que la permutación que se esta revisando en cada posición no concuerde con alguna posición de los cuadros inutilizables, esto a través de las coordenadas de estos cuadros que se guardaron; estas dos funciones devuelven un valor de verdad y luego a través de un ciclo se cuenta cuantas de estas permutaciones fueron correctas y este es el número que se imprime.

3.5 la complejidad es de $O(n!^2 * m)$.

3.6 n es el tamaño que nos están dando, esto se expande al tamaño del tablero y la cantidad de reinas a ubicar y m es la cantidad de casos que nos dan.

4) Simulacro de Parcial

4.1
4.4.1_actual > máximo
4.1.2_ $O(n^2)$ n el tamaño del arreglo

4.3.1_ i-j
4.3.2_ n
4.3.3_ O(n*m)

4.4.1_ temp%10
4.4.2_ b $O([N-M]*\log_{10}M)$

4.5.1_ i+1
4.5.2_ right==left

5) Lectura recomendada (opcional)

Mapa conceptual

6) Trabajo en Equipo y Progreso Gradual (Opcional)

6.1 *Actas de reunión*
6.2 *El reporte de cambios en el código*
6.3 *El reporte de cambios del informe de laboratorio*

PhD. Mauricio Toro Bermúdez
Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473