INFORME PROYECTO FINAL

FUNDAMENTOS DE ANÁLISIS Y DISEÑO DE ALGORITMOS

DANIEL BARRERA DEVIA

DOCENTE:

MAURICIO LÓPEZ BENÍTEZ

UNIVERSIDAD DEL VALLE

SEDE: TULUÁ

29 SEPTIEMBRE 2020

1. **Solución Ingenua**

   Para la solución ingenua se utilizó una pila, que será la estructura auxiliar que guardara todos los países al iniciar esta solución, también se usara una matriz auxiliar de las mismas dimensiones de la matriz de entrada pero con todos sus elementos en 0.

   el proceso de esta solución consta de llenar la pila con las n selecciones de manera descendente de manera que la última en ingresar sea la selección que se ubica en la posición 0,0 de la matriz de entrada.

   se comienza a evaluar la matriz de manera iterativa mientras nuestra pila auxiliar contenga elementos, mientras esta respuesta sea positiva, emplearemos otro ciclo que comenzara a partir de la posición del elemento a evaluar en filas y columnas, es decir el elemento 0, arrancara dicho ciclo desde 0,0. el 1 desde 1,1 y así sucesivamente esto debido a que evaluar las posiciones anteriores al mismo implican cálculos innecesarios puesto que en la iteración anterior ya fueron procesadas como punto principal.

   dentro de este ciclo, se evaluara si nuestra matriz principal en la posición (objetivo, i) (donde objetivo es el elemento de la pila que se está evaluando), si este es igual a 0, ingresare un 1 en la matriz auxiliar que será la que consultaremos para las respuestas finales, como ya quedo agregado ese elemento a la matriz de respuesta, debemos entonces asegurarnos que no sea accesible para las demás selecciones que tienen buenas relaciones, para ello empleo otro ciclo que llenara de valores 1 la columna y fila i,(donde i es el elemento que simpatiza con el objetivo) con esto aseguro que no sea incluido en otra villa.

   una vez se ha recorrido toda la pila ya tengo mi matriz auxiliar la que luego recorreré para extraer los elementos donde la fila donde no encuentre ningún 1, se ignora, caso contrario donde encuentre el primer uno en la fila sabrá que es una villa y la marcara en la posición (villa, selección).

   esta solución tendrá una complejidad O(n^3) en su peor caso esto debido a que se usan 3 ciclos anidados, no obstante debido a que los ciclos interiores no inician desde 0, sino que por cada iteración externa se hará una iteración menos de manera interna, el costo tiende a ser mucho menor que el mencionado, también cabe agregar que se puede dar el caso de que desde un punto determinado, el algoritmo tendrá una complejidad O(n^2)puesto que se dará que ya todas las selecciones habrán sido ubicadas.

2. Solución Voraz

Para la solución voraz se utilizó una estructura auxiliar mixta, la cual es un arreglo de pilas, donde el tamaño del arreglo será igual al número de selecciones, esto para el caso de cada región requiera una villa aparte.
lo que se hace es recorrer la matriz de entrada, teniendo en cuenta solamente su diagonal superior, puesto que al ser una matriz simétrica hacer cálculos en la otra parte de la matriz no tendría sentido, conforme se va recorriendo la matriz se valida que cada intersección fila, columna sea = 0, una vez se encuentra un 0 se empezara a evaluar si es óptimo agregarla a la solución o no, esto se realiza evaluando si la posición que es mi candidata, tiene conflicto con las que ya se agregaron a la solución, esto se realiza validando si la solución candidata es uno en la intersección de alguna de las selecciones que ya se agregaron a la estructura de las villas.

Si la candidata tiene problema con alguna de las soluciones que ya está agregada, se interrumpe el ciclo y se evalúa la siguiente columna.

De ser contrario, se agregara al arreglo en la posición del i que se está evaluando, el elemento j es decir el cruce de la fila con la columna, y automáticamente toda la columna se convierte en uno a partir de i+1 esto con el fin de que para los próximos j, se pueda evaluar la fila i para saber si tienen conflicto.

Finalmente, se crea la matriz de villas x selecciones donde el entero villas se calculara al recorrer el arreglo y contar las pilas no vacías que hay en cada posición del mismo, luego dicha matriz se rellena de unos en el cruce de las villas con la selección correspondiente.

Su complejidad será O(n^3), porque recorrer la matriz requiere un costo de O(n^2), al evaluar las relaciones que tiene mi candidato a la solución, ese recorrido me va generar un costo computacional de n, al ser un ciclo anidado, este se multiplica con el costo anterior, para finalmente dar O(n^3).

3. **Solución Dinámica**

Para la solución dinámica, se usó un arreglo de N posiciones, donde N será representado por cada selección candidata a la espera de tener asignada una villa, este arreglo en principio va a estar lleno de 0, a medida que se van evaluando las selecciones, se ira llenando con el número de villa a la que será asignada cada selección.

También se usara un Vector que me servirá como memoria temporal, donde se irán guardando las selecciones que son candidatas a ir en la villa que este evaluando, para elegir cual selección o cual selección debe ir en ese vector, lo que se hace es evaluar el primer elemento, el primer elemento al ser candidato ira a la memoria temporal, empiezo a preguntar a las demás selecciones si son amigas de ese elemento que está en memoria temporal, si es amiga, lo que hago es ingresarla a esa memoria temporal, una vez haya evaluado, todas las selecciones, mi solución será asignar a la villa actual, las selecciones que están en memoria temporal, una vez haberles asignado la villa, la memoria temporal volverá a estar vacía y preguntará a la siguiente selección del Arreglo si ya ha sido asignada una villa, si no es así irá a memoria temporal hasta que se evalúen las demás selecciones que podrían o no ser amigas, al haber evaluado todas las selecciones se asignará a la siguiente villa las selecciones que están en memoria temporal, este proceso se repite hasta haber asignado villas a cada selección.

Como resultado se tendrá un arreglo, donde indica a que villa debe ir cada selección, como por ejemplo la selección A y C, deben ir a la villa 1.

Esta solución tendrá complejidad O(n^2), el costo computacional al recorrer el arreglo será de n, al momento de ingresar un elemento en la memoria temporal y recorrer esa memoria para saber si puede ir o no con los demás elementos en esa memoria temporal, tendrá un costo computacional de n, al ingresar los datos de la memoria temporal al arreglo, este será lineal, porque se asignara uno a uno sin hacer comparaciones, finalmente al ser ciclos anidados tendrá complejidad O(n^2).