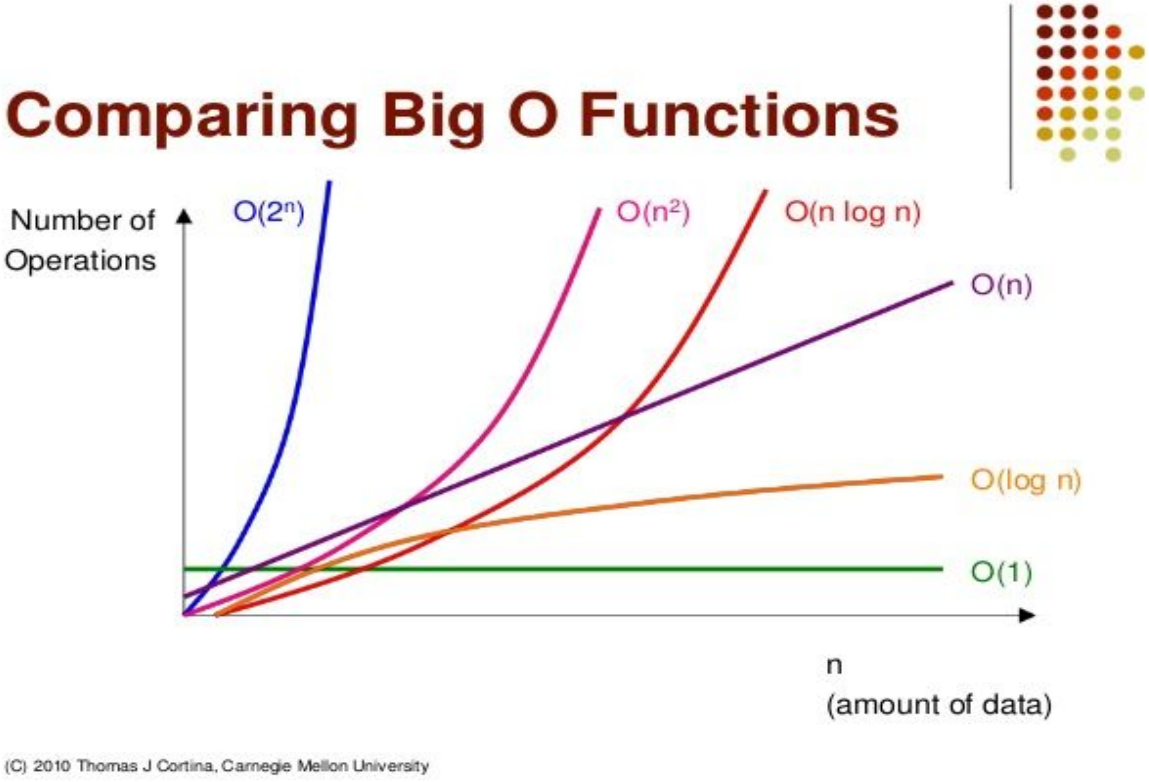
Facultad de Informática
UNIVERSIDAD NACIONAL DEL COMAHUE

Análisis de Algoritmos

Informe del trabajo práctico para promoción.



Osycka, Líam.

Legajo: FAI-1891.

29/11/2019

Parte 1

a) Si, existen distintos tipos de soluciones.
Cada una puede depender de:
   - Que tipo de implementación de grafo se use, si es con “Listas de Adyacencia” o con “Matriz de Adyacencia”.
   - Que tipo de recorrido se use para visitar los nodos del grafo, “en anchura” o “en profundidad”.

   También puede intentar resolverse desde otra perspectiva, utilizando el famoso “Algoritmo de Dijkstra”. En la solución implementada para este trabajo, se utilizó un grafo implementado con “Listas de Adyacencia” y recorriendo “en profundidad”. Cuando se quería buscar el camino más corto entre Ushuaia y Cartagena nos parabamos en Ushuaia y recorriendo el grafo íbamos obteniendo caminos posibles con sus respectivas distancias y determinando cuál era el camino más óptimo que llegara a Cartagena pasando por las ciudades obligatorias que solicitaba. En cambio, si se hubiese utilizado el ya mencionado “Algoritmo de Dijkstra” el enfoque iba a ser distinto, ya que este algoritmo calcula el camino óptimo desde el origen hacia todos los nodos, por lo que de haber sido utilizado, la implementación hubiese sido diferente.

b) En esta situación, estamos hablando de un algoritmo del tipo “Fuerza Bruta” (en el peor caso), ya que es aquel que considera como candidato a todos los elementos de, por ejemplo, la estructura con la que se está trabajando. Esto quiere decir que en el caso del problema planteado, cuando visitemos el nodo de Ushuaia hay que recorrer todos los arcos de ese nodo, y por cada nodo, todos sus arcos, y así sucesivamente hasta encontrar el camino con menor distancia que pase por todos los nodos obligatorios y llegue a destino.
   Cabe mencionar que el problema del viajante que se está presentando es del tipo NP-completo, lo que significa que a medida que se va aumentando la entrada, el costo para poder encontrar la solución óptima crece de manera exponencial.

CANTIDAD DE REPETICIONES	PARTE 1 (TIEMPO EN MS)	PARTE 2 (TIEMPO EN MS)
1000	4122	5
10000	4129	11
100000	4165	45

Parte 3

En la solución lograda en la parte 2 (dinámica) se trata de evitar llamar al método recursivo que realiza el recorrido en profundidad, para ello, cuando se consulta por la distancia del camino más corto entre 2 ciudades, primero se visita la matriz para ver si ya hay una entrada validada [1] , es decir que haya sido insertado luego de haber encontrado un camino que llegue al destino pasando por todos los nodos obligatorios. De no ser el caso, hay que invocar al método recursivo para recorrer el grafo en profundidad.
Aquí es cuando viene la gran diferencia, cuando se recorra el grafo en la implementación dinámica, se irán guardando en la matriz las distancias entre los caminos, así cuando se solicite nuevamente el camino más corto entre dos nodos, esté disponible ese valor en la matriz y no sea necesario ir al método recursivo y calcularlo de nuevo, que es lo que pasa con la solución de la parte 1, sin importar cual sea origen y destino, siempre se deberá recorrer el grafo en profundidad para encontrar el camino óptimo.

CONCLUSIONES

La solución dinámica es mejor cuando se realizan sucesivos cálculos de caminos más cortos entre varios nodos, ya que si bien hay veces que habrá que recorrer el grafo con el método recursivo, luego de determinadas iteraciones ya todos las distancias estarán en la matriz, y el coste de obtener la distancia del camino más corto se reducirá a acceder a la matriz. En este problema particular, como se

[1] Como se trabaja con un grafo completo, cuando se crean los arcos, van a haber inserciones entre todos los nodos, esto hará que haya un valor de distancia en la matriz para todos los pares de nodos, sin embargo, estaría mal utilizar este valor cuando se consulta por la matriz antes de la llamada recursiva, ya que no representa la distancia de un camino que haya pasado por todas las ciudades obligatorias.

obliga a que el camino pase por determinadas ciudades obligatoriamente y se lo solicita una única vez , con cualquiera de las dos soluciones se obtendrá el mismo resultado ya que será necesario recorrer el grafo en profundidad en un principio para identificar y obtener dicho camino.
Respecto a los tiempos, investigando, como el problema planteado es un caso particular del conocido “Traveling Salesman” hablaré de los tiempos de este. Resolviendo este famoso problema ocurrirá que el orden será del tipo O(n!), ya que por cada nodo hay que ir probando todas las posibles combinaciones, esto produce que por cada nuevo nodo que se agregue se aumenten en gran manera las llamadas recursivas. La programación dinámica con el planteo de no recalcular cosas que ya se habían hecho logran transformar esta solución en algo que se asemeja a O(2^n*n^2) , logrando principalmente, minimizar la cantidad de llamadas recursivas necesarias.