

Laboratorio Nro. 5: Programación Dinámica

Ricardo Rafael Azopardo Cadenas

Universidad Eafit
Medellín, Colombia
rrazopardc@eafit.edu.co

Jhon Jairo Chavarria Gaviria

Universidad Eafit
Medellín, Colombia
jjchavarrg@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

1. El problema del numeral 1.2 resuelve el tsp utilizando el algoritmo de held karp el cual tiene tres pasos fundamentales para hallar el costo mínimo:
 - Lo primero es generar todos los subconjuntos de todos los vértices a todos los vértices excluyendo el vértice inicial, en la implementación los conjuntos fueron representados por bitmaskSets, los cuales funcionan como una implementación más eficiente de Sets de java.
 - Luego, se calcula la distancia desde el nodo inicial o nodo 0 hasta cada uno de estos conjunto y consecuentemente se calcula la distancia de estos conjuntos a los otros conjuntos, estos valores se van grabando en una tabla para ser consultados y comparados.
 - Por último, se elige la distancia mínima entre cada uno de estos nodos haciendo uso de la tabla y se elige desde cual vértice final se logra establecer la conexión con costo mínimo nuevamente hacia el vértice inicial.
2. Fuera de la solución del vecino más cercano que ya conocemos y la solución por programación dinámica que no admite un conjunto de datos tan extenso podemos encontrar otras técnicas aproximadas que aplican heurísticas para acercarse a una solución más precisa, tal es el caso del A*, el cual aplica una heurística para determinar una distancia de ida y vuelta desde un nodo inicial hasta un nodo final.
3. El algoritmo utilizado para resolver el ejercicio en línea del numeral 2.1 es el mismo Held Karp, es posible utilizar este algoritmo si se reduce el mundo de Karolina un grafo completo que solo tenga la posición de ella y la de los residuos radioactivos, el grafo completo se reduce de tal forma que los pesos entre todos los vértices se escriben con respecto a los movimientos en los

DOCENTE MAURICIO TORO BERMÚDEZ

Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627

Correo: mtorobe@eafit.edu.co

que solamente Karolina se puede mover, de tal forma que el grafo generado es posible aplicarle un Held Karp con posición inicial de Karolina y se devuelve a la posición inicial luego de visitar todos los otros vértices (desechos radioactivos), efectivamente encontrando el costo mínimo.

4. La complejidad del algoritmo de Held Karp es de $O(2^n * n^2)$.
5. N es la cantidad por la cual las operaciones están en función, en este caso está definido por la entrada, específicamente el número de conexiones en el grafo completo.

4) Simulacro de Parcial

1.1

		C	A	L	L	E
	0	1	2	3	4	5
C	1	0	1	2	3	4
A	2	1	0	1	2	3
S	3	2	1	1	2	3
A	4	3	2	2	2	3

1.2

		M	A	D	R	E
	0	1	2	3	4	5
M	1	0	1	2	3	4
A	2	1	0	1	2	3
M	3	2	1	1	2	3
A	4	3	2	2	2	3

2.1. $O(n*m)$

2.2. `return table[i-1][j-1];`

3.1. $O(n)$

3.2. $T(n) = c1:n + c2$

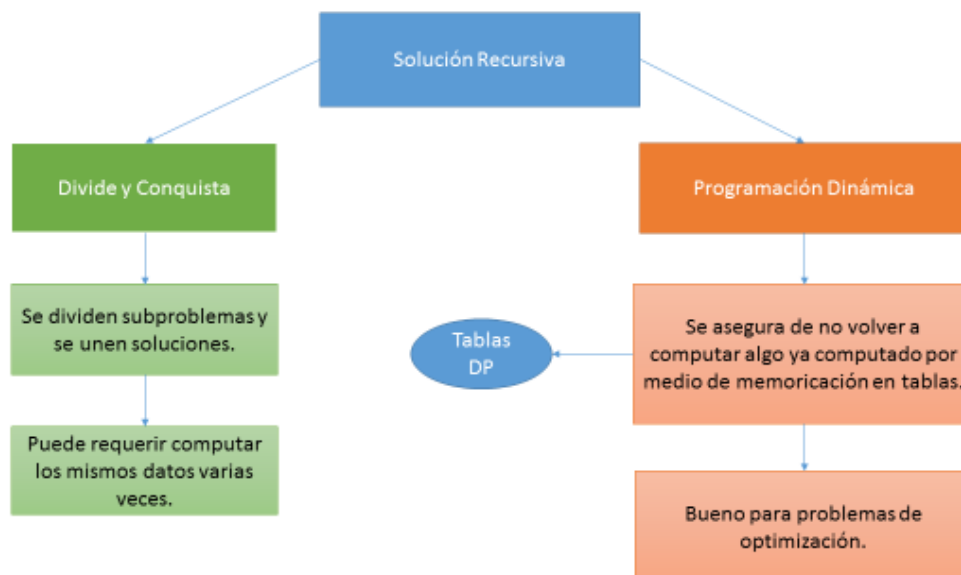
4. Ninguna de las anteriores.

5. Lectura recomendada (opcional)

Resumen: Programación Dinámica

Muchas veces cuando programamos se nos presentan una serie de problemas en la cual su solución viene de una sucesión secuencial de repetir los mismos pasos una y otra vez, es decir, su solución se alcanza de forma recursiva, incluso en muchos problemas podríamos alcanzar una solución eficiente utilizando métodos recursivos en muchas técnicas, por ejemplo, en divide y conquista, que dividimos el problema en subproblemas más pequeños, calculamos sus soluciones recursivamente y luego juntamos las soluciones.

Sin embargo, existe un problema cuando para computar estos subproblemas necesitamos información computada anteriormente, por lo tanto calcular estas soluciones se vuelven gradualmente recursivamente inalcanzables. Una solución a este problema es la programación dinámica, la cual sigue un principio muy importante: Algo ya computado no lo computaras dos veces. Lo logra de la siguiente manera, las soluciones que se van encontrando se pueden ir guardando en una tabla, de tal forma que no sea necesario volver a generar sus soluciones, sino que simplemente se pueden consultar su valor. De esta forma la programación dinámica nos ayuda a optimizar muchos problemas, que normalmente solo operarían para un rango pequeño de datos por métodos recursivos convencionales.



6. Trabajo en Equipo y Progreso Gradual (Opcional)

Actas de Reunión:

Integrante	Fecha	Hecho	Haciendo	Por Hacer
Ricardo Azopardo & Jhon Chavarria	17/10/2017	Implementación de algoritmos de levenshtein y subsecuencia más larga.	Held Karp.	Terminar Held Karp. Ejercicio en Línea. Informe
Ricardo Azopardo	19/10/2017	Avances en Held Karp e Informe.	Held Karp.	Terminar Held Karp. Ejercicio en Línea. Informe.
Ricardo Azopardo	20/10/2017	Complejidad de los algoritmos e Informe.		Lecturas.