PECAS

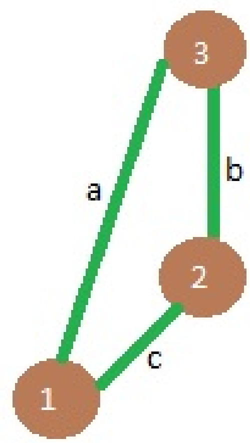
Sara Chica, Rodrigo Gualtero

20 de Septiembre, 2012

Índice

1. Introducción

Este es un problema de la UVA, identificado con el código *10034*, en el cual Richie debe encontrar como conectar todas las pecas de la espalda de su padre con la menor distancia posible, para de esta forma reducir la cantidad de tinta que se usa para unirlas.

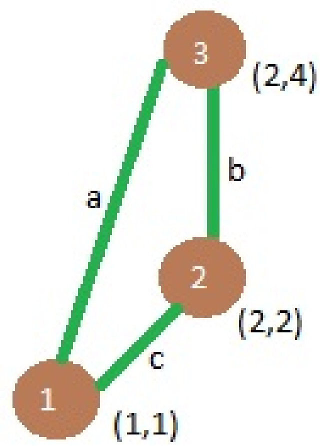


Ejemplo 1: 3 Pecas

En este ejemplo las pecas son *1*, *2* y *3*; y las posibles conexiones son a, b y c.
Utilizando algunas de estas 3 conexiones Richie debe poder conectar todas las pecas con la menor distancia.

2. Definición del problema

Para poder obtener la distancia entre peca y peca se hace necesario tomar cada peca como un punto ubicado en el plano cartesiano, por lo cual tendría componente x y y.



Ejemplo 2: 3 Pecas En plano cartesiano

En el *ejemplo 2* se pueden observar las mismas 3 pecas que en el *ejemplo 1*, pero ahora ubicadas en el plano cartesiano, cada una con sus respectivas cordenadas.

2.1. Entrada

Como primer lugar entra un valor que determina el número de nodos, este debe ser entre 0 y 200.
Después, separado por una linea en blanco, entra un número, el cual indica la cantidad de pecas.
Finalmente entran todas las pecas con formato «x y»; donde x y y pueden ser números reales.

2.2. Salida

Imprime la distancia más corta para conectar todas las pecas.

3. Modelamiento matemático

Sea G un grafo, $G(V, L)$, donde

V es el conjunto de nodos del grafo,

$$V = \{a_1, a_2, ..., a_n\}$$

y L es el conjunto de todas las conexiones,

$$L = \{(a_1, a_2), (a_1, a_3), ..., (a_1, a_n), (a_2, a_3), (a_2, a_4), ..., (a_{n-1}, a_n)\}$$

Este grafo cumple con las siguientes propiedades:

1. El grafo es conexo, es decir existe una sucesión de arcos tal que es posible encontrar un camino para llegar a cualquier elemento del Conjunto.
2. El grafo es no dirigido, es decir $(a, b) = (b, a)$.
3. Ningun nodo tiene conexión consigo mismo, es decir no existe una conexión tal que (a, a); es decir $(a, a) \notin L$

4. Planteamiento de la Solución

Para solucionar el problema se debe usar un algoritmo que permita encontrar el camino mas corto, para ello existen ya algunos, como lo son:

1. Algoritmo de Dijkstra: Encuentra el camino más corto desde un único nodo origen, hasta todos los otros del grafo.
2. Algoritmo de Bellman-Ford: Encuentra el camino más corto desde un nodo origen, permitiendo que el peso de algún arco sea negativo.
3. Algoritmo de Floyd-Warshall: Encuentra el camino más corto entre todos los nodos del grafo.
4. Algoritmo de Kruskal: Encuentra el camino más corto el cual pueda conectar todos los nodos, sin necesidad de tener un nodo origen.

Por las características del problema se hace más conveniente utilizar el algoritmo de Kruskal.
Lo que realiza este algoritmo como primera medida es organizar el arreglo de acuerdo a los pesos que tenga cada arco; luego de esto comienza a mirar las diferentes conexiones para hallar el camino más corto por medio del cual se puedan conectar todos los nodos.

5. Conclusiones

1. Entre los algoritmos consultados el que permite resolver el problema de una forma más eficiente es el de Kruskal, ya que, como primera medida no necesita un nodo de origen, y además de esto, está hecho para resolver problemas de este tipo.

2. Existen varias implementaciones del algoritmo, por lo cual es importante cerciorarse si el algoritmo implementa el ordenamiento del arreglo, en caso de que no, se hace necesario pasarle este ordenado de peso menor a mayor.

3. Estos algoritmos que resuelven problemas en donde se debe encontrar el camino más corto, tienen aplicaciones en redes y telecomunicaciones, facilitan el diseño de sistemas, entre otros.