
Algorithms for Problem Solving – 11650

Algoritmos sobre grafos

Jon Ander Gómez Adrián
jon@dsic.upv.es

Departament de Sistemes Informàtics i Computació
Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

11 de marzo de 2014

111001/10034

- Precalculamos una matriz de distancias de todos a todos. Utilizamos la distancia Euclídea, incluida la raíz cuadrada, ya que nos hace falta para obtener el valor final.
- Aplicamos el algoritmo de Prim desde el primer nodo.
- Recorremos todos los caminos desde cada nodo al inicial gracias al árbol guardado en el vector que nos dice que nodo precede a otro.
- Acumulamos y seguimos aquellos caminos de vuelta mientras un nodo todavía no ha sido pintado.

111002/10054

- En este problema es mejor mantener una única lista de arcos, si es doblemente enlazada mejor, así el coste de quitar un arco es $\theta(1)$.
- Buscamos piezas y las vamos colocando hasta completar un ciclo o hasta que no queden piezas que encajen.
- Los trozos que hemos construido, sean ciclos o no, comprobamos que los podamos encajar todos. Ojo, no tiene porqué ser al principio o al final.
- Si todos los trozos los hemos podido fundir tenemos solución, si queda algun trozo suelto diremos *“some beads may be lost”*.
- Este problema se puede atacar desde otra perspectiva, por ejemplo intentando completar un camino de Euler. Si encontramos uno que abarca todos los arcos ya tenemos solución.

111003/10278

- Con el algoritmo de Prim aplicado desde cada nodo que tiene una estación de bomberos obtenemos la distancia de cada nodo que no tiene estación al más cercano que tiene una.
- Después buscamos el nodo sin estación tal que si la colocamos en él minimiza la distancia global de nodos a estaciones.
- Para ello aplicamos de nuevo Prim a partir de cada nodo candidato.
- Aquél que reduce la máxima distancia a estaciones de los restantes es el candidato.

111004/10039

- Este problema yo lo he resuelto mediante programación dinámica.
- Descartar todos los trenes cuya hora de llegada es anterior a la de salida. Son los que llegan al día siguiente de haber partido.
- Preparamos un array bidimensional de tantas filas como minutos tiene un día. O para acabar antes, de 2400, lo que representa 24 horas de 100 minutos. Se consume más memoria pero resulta más sencilla la solución.
- Desde cada hora y ciudad alcanzadas se intenta alcanzar aquellas otras compatibles según las restricciones de tiempo.
- Cuando se vaya a procesar la ciudad destino podemos finalizar.
- Si la ciudad destino no ha podido ser alcanzada diremos “*No connection*”.

111007/10249

- Ordenamos los equipos en orden descendente por número de miembros.
- Por cada equipo:
 - Reordenamos las mesas en orden descendente por número de asientos libres.
 - Asignamos los miembros del equipo en las mesas con sitios libres a razón de uno por mesa.
- Si nos sobran personas de un equipo ya podemos decir que no los podemos repartir según las restricciones.
- Si llegamos al final es que los hemos podido sentar a todos y tenemos solución.

Resumen problemas

Freckles
The Necklace
Fire Station
Railroads
The Grand Dinner
Resumen
▷ problemas

- 111001/10034 “Freckles”
- 111002/10054 “The Necklace”
- 111003/10278 “Fire Station”
- 111004/10039 “Railroads”
- 111007/10249 “The Grand Dinner”