Algoritmos Greedy

DGIIM 2017 - Grupo 2

Jesús Sánchez de Lechina Tejada
Álvaro López Jiménez
Miguel Ángel Robles Urquiza
Antonio Martín Ruiz

Índice

Algoritmos Greedy

1. Análisis del problema

Problema: encontrar un recubrimiento para un grafo dado.

Recubrimiento: conjunto de nodos tal que todos los nodos del grafo pertenecen al conjunto, o son adyacentes a alguno del conjunto.

Optimalidad: Encontrar un recubrimiento con el menor número de vértices posible.

2. Diseño de la solución.

- ***Lista de candidatos:*** Todos los nodos del grafo

- ***Lista de candidatos utilizados:*** Nodos seleccionados del grafo original y los nodos adyacentes a él.

- ***Función solución:*** La lista de candidatos utilizados coincide con la lista de candidatos, es decir, los nodos recubiertos son todos los nodos del grafo.

- ***Función selección:*** Nodo con mayor número de adyacencias que no esté en el recubrimiento.

- ***Criterio de factibilidad:*** El nodo que seleccionamos no está en el recubrimiento.

- ***Función objetivo:*** Obtener un recubrimiento con el mínimo de nodos.

3. Esqueleto del algoritmo

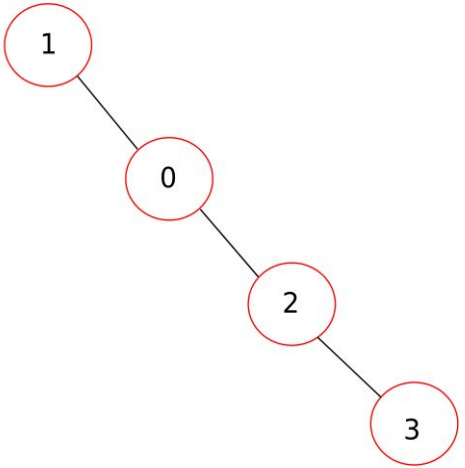
```
vector<int> recubrimiento ( vector<vector<int> >  m) {
        S = solución = conjunto de nodos que conforman el recubrimiento
        N = lista de nodos
        NC = lista de nodos cubiertos por el recubrimiento
        LCU = lista de nodos utilizados

        Mientras ( NC != N )  {
                nodo = nodo con mayor nº de incidencias no utilizado = getNodoMaxInc(m,LCU)
                Si el nodo no está ya en el recubrimiento {
                        NC <- nodo
                        NC <- nodos adyacentes a nodo = adyacencias (nodo, NC, m)
                        S <- nodo
                }
        }
        devuelve S
}
```

4. Ejemplo de funcionamiento

Sea el grafo de la figura y su matriz de adyacencia, el resultado de nuestro algoritmo sería:

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



Recubrimiento formado por los nodos:

0 2

```
usuario@usuario-W330AU:~/Escritorio/directoriogit/PracticasAlgor-tmica/Práctica
3$ ./recubrimiento recubrimiento.dat
TAM: 4

0110
1000
1001
0010


Solución = Recubrimiento formado por los nodos 0 2
Coste = 2
```

5. Aplicaciones

Existen varias aplicaciones del recubrimiento de grafos, las más comunes son algo más abstractas, aplicaciones en demostraciones de teoría de grafos.

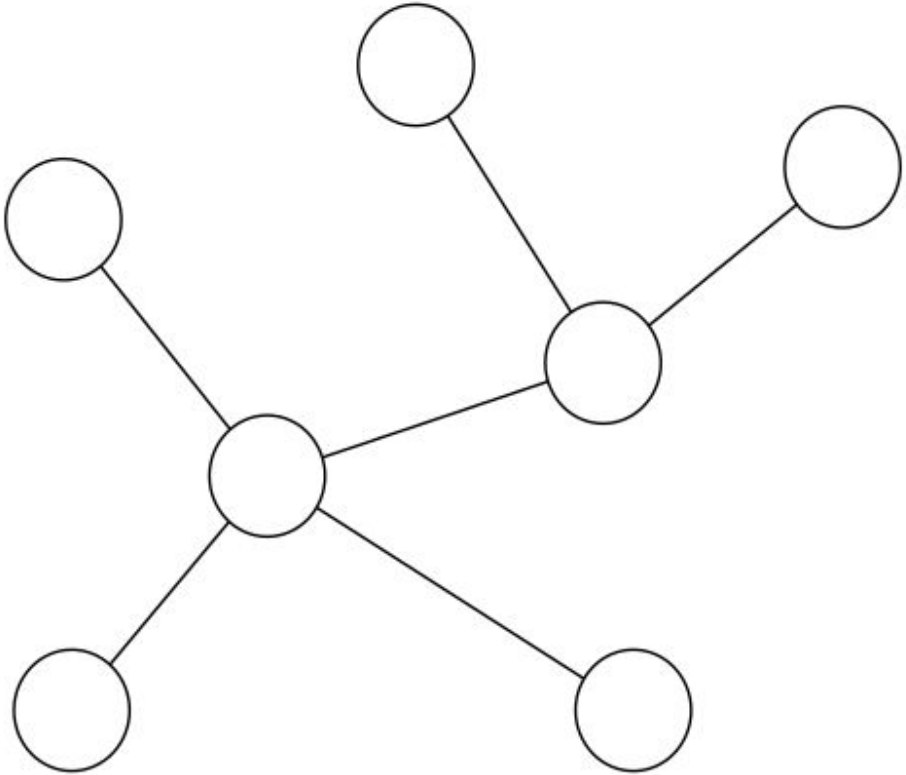
Pero podemos usar otras aplicaciones algo más inusuales, a expensas de no ser la alternativa más eficiente. Un ejemplo de esta es "*El problema de Félix Rodríguez de la Fuente*"

El problema de Félix Rodríguez de la Fuente

El amante de los animales Félix es el encargado de alimentar a una población de patos.

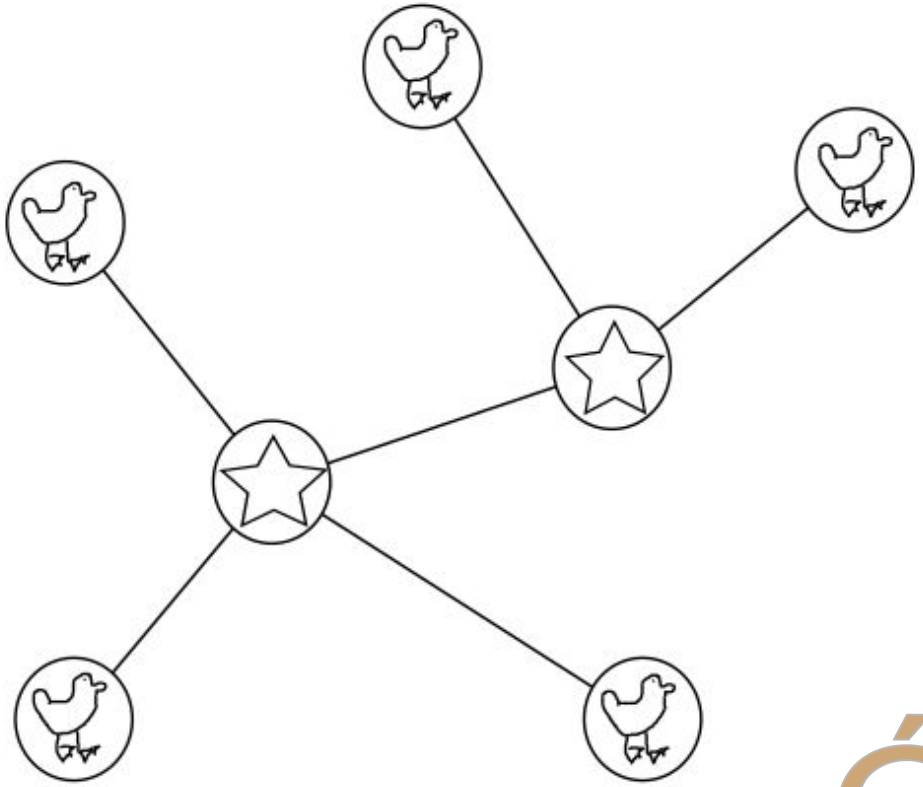
Estos se encuentran divididos en distintos hábitats que están unidos por pasos aviarios (*pasos de patones*) por los que pueden pasar. Pero estos animales en cautividad sólo pueden desplazarse a un hábitat adyacente al día y tienen que estar de vuelta a la noche para dormir en sus nidos.

Félix debe de asegurarse de establecer comederos para que todos los patitos estén bien alimentados.



Hábitat

Paso aviario



Óptimo

6. Órdenes de eficiencia

El orden de eficiencia del algoritmo en el peor de los casos es: **$O(n^3)$**

En recubrimiento podemos ver un bucle for, pero este no resulta ser el más relevante desde el punto de vista computacional. Inmediatamente después nos encontramos un bucle while que, en el peor de los casos tiene que hacer n iteraciones, pero en cada iteración está llamando a otras funciones con bucles en su interior. Las llamadas más costosas tienen dos bucles for anidados. Por lo que este problema es de orden $O(n^3)$.

7. Compilación y más ejemplos de ejecuciones

- Para compilar este proyecto tenemos que usar el siguiente comando en la terminal:

  prompt> g++ -o recubrimiento recubrimiento.cpp

  o bien ejecutar *make* en la terminal.

- Para ejecutar el proyecto debemos usar el siguiente comando:

  prompt> ./recubrimiento recubrimiento.dat

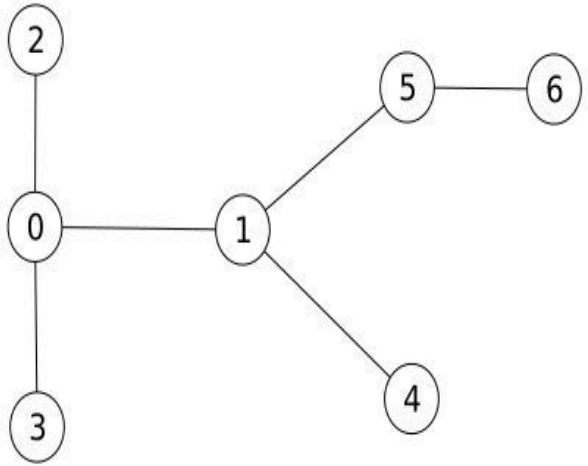
  donde recubrimiento.dat es un archivo de texto que contiene la matriz de adyacencia del grafo con el formato adecuado.

7. Ejemplos de ejecuciones

```
usuario@usuario-W330AU:~/Escritorio/directoriogit/PracticasAlgor-tmica/Práctica
3$ ./recubrimiento recubrimiento3.dat
TAM: 7

0111000
1000110
1000000
1000000
0100000
0100001
0000010


Solución = Recubrimiento formado por los nodos 0 1 5
Coste = 3
```



7. Ejemplos de ejecuciones

```
usuario@usuario-W330AU:~/Escritorio/directoriogit/PracticasAlgor-tmica/Práctica
3$ ./recubrimiento recubrimiento5.dat
TAM: 6

011000
101100
110010
010011
001100
000100


Solución = Recubrimiento formado por los nodos 1 2 3
Coste = 3
```