Parte I Problema 3 - La comunidad del anillo

1.1 Descripción del problema a resolver

La empresa `AlgoNET` nos contrató para que le brindemos una solución algorítmica a un problema de redes. Nuestro cliente quiere ofrecer un sevicio particular sobre una red existente de computadoras. La red cuenta con un conjunto de conexiones entre pares de computadoras, donde cada conexión tiene un costo asociado. El objetivo es elegir un subconjunto de conexiones y de equipos (que funcionarán como servidores) tales que:

1. El conjunto de servidores deberá conformar un anillo
2. Todo equipo debe quedar conectado al anillo de servidores vía un enlace directo o pasando a través de otros equipos en el medio
3. La suma total de los costos de todos los enlaces utilizados debe ser lomínimo posible
4. El algoritmo debe tener una complejidad estrictamente mejor que $O(n^3)$
5. El algoritmo debe detectar los casos en los que no haya solución

El formato de entrada contiene una instancia del problema. La primera línea contiene un entero positivo n que indica la cantidad de equipos de la red (numerados de 1 a n), y un entero no negativo m que corresponde a la cantidad de enlaces disponibles. A esta línea le siguen m líneas, una para cada enlace, con el formato `e1 e2 c` donde `e1` y `e2` representan los equipos en lso extremos del enlace en cuestión (ambos enteros entre 1 y n) y `c` es el costo por utilizar dicho enlace. En caso de haber solución, la salida tiene el formato `C Ea Er` donde `C` es el csoto de la solución dada y `Ea` y `Er` son los enlaces utilizados por el anillo y para el resto de la red, respectivamente. A esta línea le siguen `Ea` líneas, una para cada enlace utilizado en el anillo y luego `Er` líneas, una para cada enlace utilizado fuera del anillo, todas con el formato `e1 e2` donde `e1` y `e2` representan los extremos del enlace en cuestión. Si hay más de una solución óptima, se puede devolver cualquier, y si no hay solución se debe devolver la palabra `no`. A continuación, mostramos un ejemplo de una posible instancia.

Ejemplo de entrada: instancia 1

```
8 10
1 2 1
2 6 1
2 3 5
2 7 2
6 7 1
7 5 3
3 5 8
5 4 2
5 8 2
4 8 10
```

Ejemplo de salida: instancia 1

```
20 3 5
2 6
6 7
2 7
1 2
7 5
3 5
5 4
5 8
```

A continuación, presentamos una representación del problema, visto como un grafo: cada nodo es un equipo y cada arista es una conexión entre algún par de computadoras, y tiene un costo asociado a dicha conexión, seguido de tres posibles formas de elegir los subconjuntos de equipos y conexiones requeridos por el problema

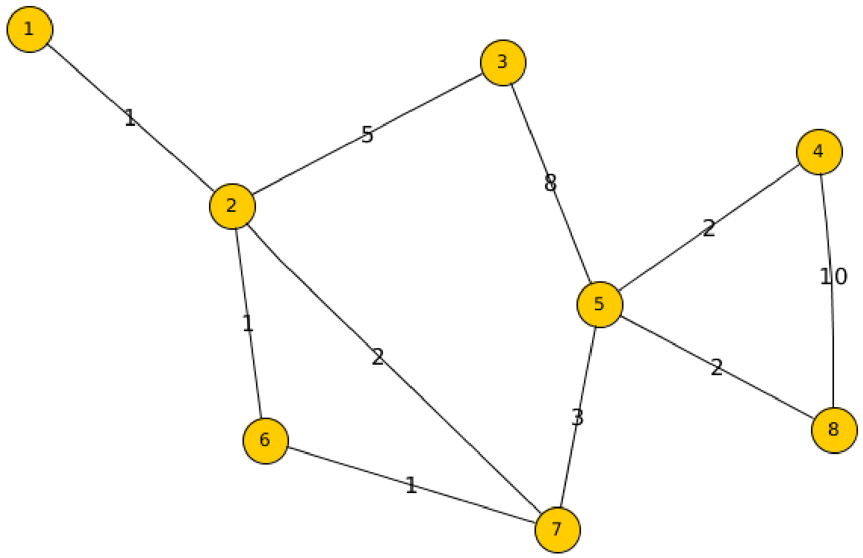

Figure 1.1.1: Red de equipos

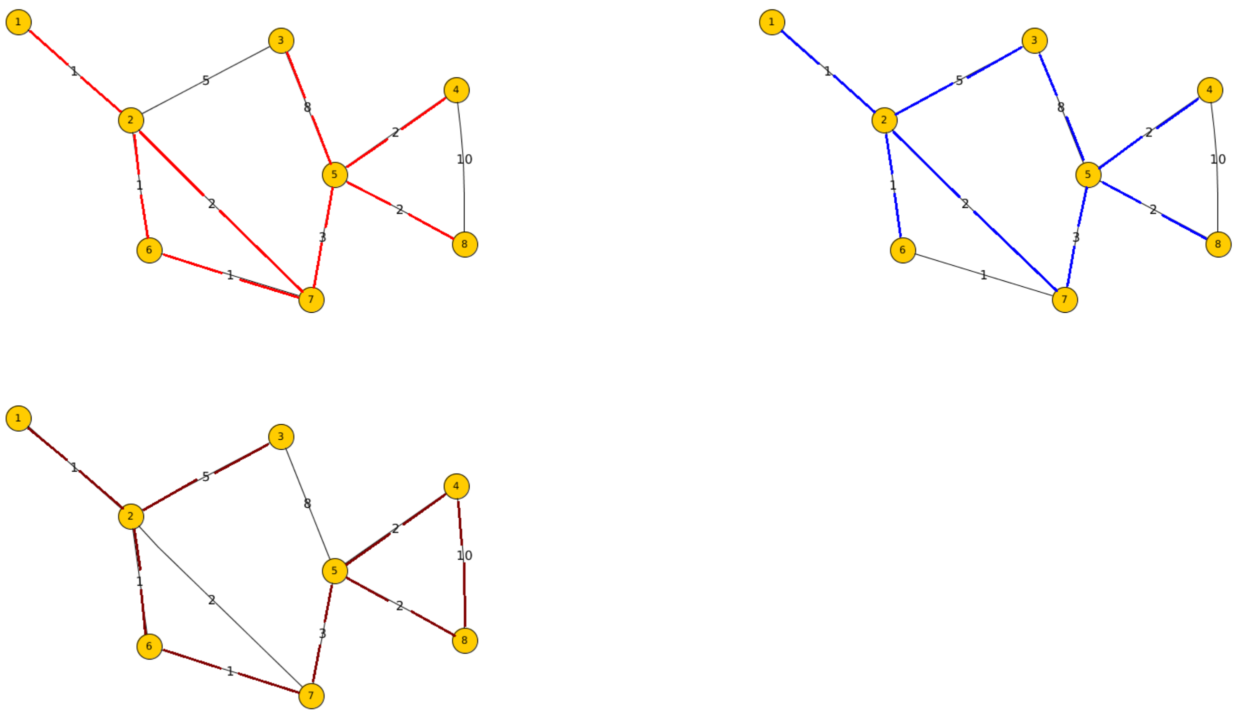

Figure 1.1.2: Soluciones de costo C = 20 (rojo), C = 24 (azul), C = 25 (marrón) respectivamente

Se puede observar que las tres soluciones presentadas en la figura 1.1.2 son efectivamente soluciones porque todas presentan algún circuito de servidores, y en todos los casos, todas las computadoras de la red están conectadas con los servidores con conexiones directas o a través de conexiones con otros equipos. Sin embargo, no todas estas soluciones son óptimas: una tiene costo 20, otra 24 y otra 25. Es decir, que para este caso en particular, nuestro algoritmo debería devolver como solución los servidores 2, 6, 7 y los enlaces que están marcados en rojo en la figura 1.1.2.

1.2 Ideas desarrolladas para la resolución

A partir del ejemplo presentado en la sección anterior y de algunos otros, observamos que en todos los casos, todos los enlaces que conformaban el árbol generador mínimo (AGM) del grafo generados por nuestra red de equipos estaban incluidos en la solución óptima del problema. En nuestro ejemplo, el AGM sería:

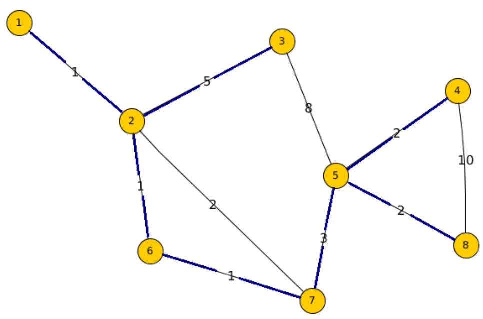

Figure 1.2.3: Árbol Generador Mínimo del grafo del ejemplo de la figura 1.1.1

Entonces, pensamos que posiblemente esto siempre sucediera, al menos siempre que existiera alguna solución. Así es que, intentamos probar que tomando el AGM del grafo en cuestión y luego agregándole la arista de menor pero de las que no estuvieran incluidas en el AGM, el grafo generado sería una solución (sería un grafo con algún ciclo y todos los nodos estarían conectados con algún nodo del conjunto de nodos que conformaban el ciclo en el grafo) y que sería óptima (la suma total de los pesos de los enlaces sería lo mínimo posible). La demostración está más adelante.
Entonces, a continuación describimos los pasos que nuestro algoritmo efectuará para resolver el problema planteado:

1. generar un grafo G en donde cada nodo sea un equipo y cada arista un enlace entre un par de equipos, con un costo asociado igual al costo que hay que pagar por el ancho de banda empleado por dicha conexión

2. chequear si el grafo G es conexo. Si no lo es, entonces el problema no tendrá solución

3. si el grafo G es conexo, generar el árbol generador mínimo T del grafo G

4. buscar el menor de los enlaces que aparece en el grafo G pero que no aparece en el árbol generador mínimo T. Llamemos a ese eje e.

5. agregamos el eje e al AGM T. Queda así formado el grafo $G' = T + e$, que es un grafo conexo y tiene un único ciclo. Además, cumple con todo lo que nos pide el enunciado del problema

6. devolvemos la suma total de los costos de todas las conexiones que son utilizadas por nuestro grafo G' y, por otro lado, devolvemos los enlaces que conforman nuestro anillo de servidores y el resto de los enlaces de nuestra solución

1.3 Demostración de Correctitud

Para demostrar la correctitud de nuestro algoritmo, tenemos que demostrar que los pasos descriptos en la sección anterior para la resolución del problema efectivamente nos conducen a la generación de una solución correcta. Entonces, repasando las ideas ya expuestas, lo que planteamos es que se puede llegar a una solución óptima a partir de generar el AGM T de nuestro grafo G y luego generar el grafo $G' = T + \{e\}$, donde e es el eje de menor costo que esta en G y no en T. Pero lo que nos pide el enunciado es que generemos, a partir del grafo G, otro grafo G'' tal que G'' sea conexo, contenga un único ciclo y que la suma total de los costos de sus ejes sea lo mínima posible. Entonces, demostrando la siguiente proposición, estaríamos demostrando nuestro método empleado para la resolución es efectivamente correcto:

Proposición 1. *Dado un grafo G conexo, con T un AGM de G y S el conjunto de todos los árboles generadores de G y e el mínimo eje perteneciente a $G \backslash T$ vale que:* $costo(T+e) = \min\limits_{\substack{f \in G \backslash T \\ T' \in S}} costo(T'+f)$

Demostración de la Proposición 1. Demostraremos ambas desigualdades:

$\leq$: en este caso, vale que $costo(T) \leq costo(T')$ y además $costo(e) <= costo(f)$. Entonces, es evidente que $costo(T+e) = costo(T) + costo(e) \leq costo(T') + costo(f) = costo(T'+f)$
$\geq$: Tomando $T' = T$ y $f = e$, tenemos que $costo(T'+f) = costo(T+e)$. Entonces, sigue que $\min\limits_{\substack{f \in G \backslash T \\ T' \in S}} costo(T'+f) \leq costo(T+e)$

□

1.4 Justificación de cota de complejidad

input : ???   **output**: ??

Algorithm 1: Pseudocódigo para el algoritmo empleado en la resolución

1.5 Testeos de complejidad