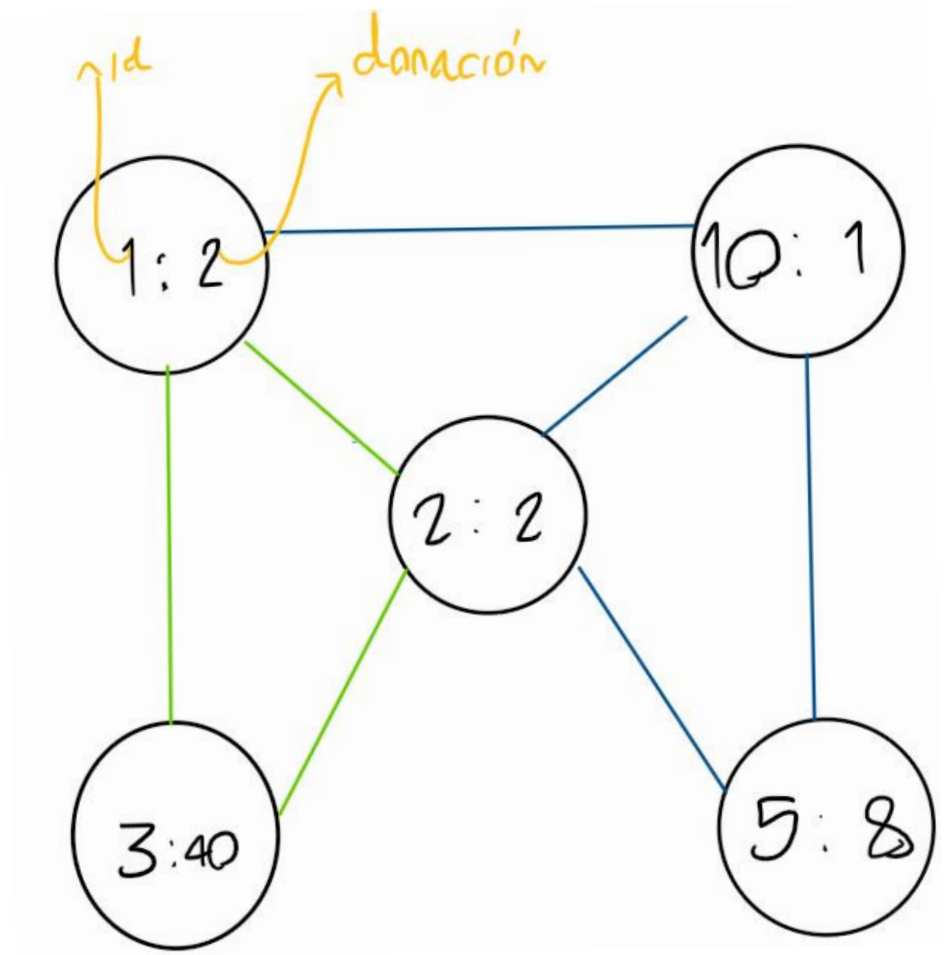
Proyecto 3

Sergio Franco Pineda (202116614), Lina María Ojeda Amaya (202112324), Germán Alberto Rojas Cetina (202013415)

1. Algoritmo de solución

1.1. Explicación del algoritmo implementado

Se empieza con un grafo, donde el objetivo es identificar el clique entre los cliques que maximiza el valor de la donación.



Por ejemplo, en este grafo la donación máxima es la aportada por el clique formado por los vértices 1, 3 y 2.

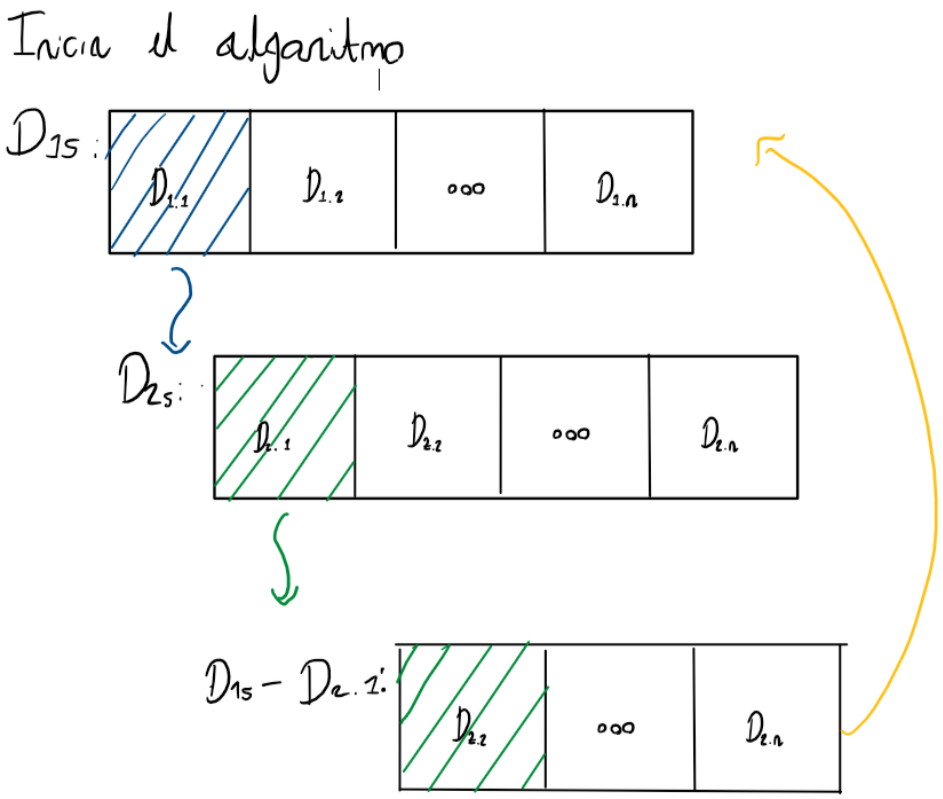
El algoritmo recorre la lista de los donantes y por cada uno, llamémosle D1s, guarda el dinero que este va a donar en "la bolsa" y recorre la lista de sus donantes conocidos, a estos llamémosles D2s.

Se recorren todos los D2s y se comprueba que cada D2 también conoce a D1, si es así se añade D2 como un donante y se añade su dinero a la bolsa junto con el de D1.

Luego, dentro del recorrido de los D2s se inicia otro recorrido sobre la lista de conocidos de D1 (la misma de donde obtuvimos D2), llamémosles D3s, y se hace lo siguiente, siempre y cuando D3 no sea igual D2, esto para garantizar que clique de donantes no esté limitado a D1, D2 y D3. Lo que se hace en este recorrido es recorrer los donantes que hemos tenemos hasta el momento para verificar que D3 conozca a todos los donantes que tenemos hasta el momento en la lista de donantes ya que todos se

tienen que conocer mutuamente, si sí lo hace, se añade D3 a la lista de donantes y su dinero a la bolsa. Esto se repite por cada D2

Al finalizar el último recorrido iniciado en el párrafo anterior (el de los D3s), se revisa si la última lista de donantes que recogimos junta una cantidad mayor a la actual, si es así, ahora esta última lista de donantes serán los donantes. Todo el proceso anterior se repite por cada D1.



1.2. Alternativas de implementación diferentes

Ya que la raíz de este problema es el problema de encontrar cliques, una alternativa sería la de fuerza bruta, es decir, encontrar todos los cliques del grafo y luego ir por cada vértice de cada donante de cada clique sumando su aporte para al final elegir el clique que tiene el máximo aporte de dinero.

1.3. ¿Por qué se escogió la alternativa implementada?

Aunque la alternativa nombrada anteriormente no es incorrecta, se escogió la alternativa implementada ya que si bien para grafos pequeños la alternativa nombrada funciona perfectamente, en grafos de tamaño significativos demoraría mucho debido a la naturalidad de su complejidad temporal.

1.4. ¿Por qué la alternativa implementada resuelve perfectamente el problema?

Es importante saber que este problema deriva del problema CLIQUE, el cual pertenece a la clase NPC y no existe algoritmo en tiempo polinomial que lo pueda resolver, es por esto que nuestro algoritmo brinda una solución 2 aproximada del problema, es decir, al tratarse de un problema de maximización nos permitirá conocer cuál es la máxima donación con una desviación negativa de máximo dos veces el valor de la donación realmente máxima; esto nos lo permite nuestra manera de recorrer el grafo y buscar las relaciones ideales entre los donantes.

Nuestra base para decir que es 2 aproximada es que entre todos los casos de prueba donde no se dio la solución real sino la aproximada, la solución aproximada se desvió máximo dos veces la solución real de esta misma.

2. Análisis de complejidad

Sea V la cantidad de donantes.

2.1. Complejidad temporal

Nuestro algoritmo realiza 4 recorridos anidados donde cada recorrido recorre máximo V donantes. Los tres primeros recorridos son los que podemos observar en la imagen de la sección 1.1 y el cuarto se ubica dentro del tercer recorrido, que es el encargado de verificar que el elemento que se tiene en el tercer recorrido conoce a todos los demás donantes que se han añadido a la lista de donantes.

Por lo tanto, nuestra complejidad temporal es de $O(V^4)$.

2.2. Complejidad espacial

Nuestro algoritmo guarda un set de donantes $O(V)$, un número para la donación máxima $O(1)$, una lista de enteros donde el elemento i es el dinero del donante i $O(V)$, y el grafo como un diccionario de adyacencia con máximo V llaves donde la llave i tiene como valor una lista de máximo $V-1$ donantes conocidos $O(V^2)$. Por lo tanto, nuestra complejidad espacial es de $O(V^2)$.

3. Compresión de problemas algorítmicos

3.1. Las personas podrían aceptar no conocer máximo una de las personas que van a aportar

3.1.1. ¿Qué nuevos retos presupone este nuevo escenario?

No presupone ningún reto ya que la modificación con respecto a la implementación original es mínima.

3.1.2. ¿Qué cambios tendríamos que realizar a nuestra solución para que se adapte a este nuevo escenario?

En el último recorrido anidado de nuestra implementación, que es donde se verifica que un donante conozca a todos los demás, tenemos una condición que interrumpe el recorrido si el donante que tenemos en el momento no conoce a alguno de los que ya están añadidos a la lista de donantes (esto es más que todo para reducir el tiempo de respuesta del algoritmo) y no se añade a la lista de donantes. Debemos no interrumpir el recorrido y más bien contar a la cantidad de donantes que sí conoce, al final, si la cantidad de donantes que conoce es mayor o igual a la cantidad de donantes que tiene la lista actual menos 1, podemos añadir al donante actual.

3.2. Las personas solamente necesitarían conocer al menos una de las personas que van a aportar

3.2.1. ¿Qué nuevos retos presupone este nuevo escenario?

No presupone ningún reto ya que la modificación con respecto a la implementación original es mínima.

3.2.2. ¿Qué cambios tendríamos que realizar a nuestra solución para que se adapte a este nuevo escenario?

La modificación es similar a la explicada en la sección 3.1.2. Debemos no interrumpir el recorrido si no conoce a algún donante y añadir si el conteo de conocidos es mayor o igual a 1.