Nombre: Christofer Rodríguez Rut: 20.239.786-7 Sección: B-2

Problema 1

Si calculamos el orden de complejidad de cada algoritmo, obtenemos lo siguiente:

a) $O(n^3)$
b) $O(n^2 * log^2(n))$
c) $O(n^3)$
d) $O(nlog(n))$
e) $O(n^2 \log(n))$

Si ordenamos las expresiones obtenidas de menor a mayor, podemos notar que el algoritmo más eficiente es el algoritmo **D**, ya que posee el orden de complejidad crece más "lento" ante el aumento del tamaño de la entrada.

Problema 2

a) No se puede obtener información sobre esta transformación, ya que no se sabe la naturaleza del problema original (**R**).
b) Podemos concluir que problema **Q** pertenece a **NP**-completo, ya que se comprueba que pertenece tanto a **NP** como a **NP**-duro, además que el problema **R** pertenece a **NP**-duro.
c) Podemos concluir que el algoritmo **R** pertenece a **P**, ya que a diferencia del caso con los **NP** donde deben cumplir ciertas características, **R** se puede transformar a **T** en un tiempo polinomial y como **T** ya es del tipo **P**, **R** también lo es por transitividad.
d) No podemos concluir nada, ya que no se sabe la naturaleza del algoritmo original (**R**), para poder concluir algo, la transformación debería haber sido al revés.

Problema 3

a)

Teniendo un grafo G(V,A) donde V son los vértices, A las aristas, m la cantidad de aristas y n la cantidad de vértices. Cada uno de estos vértices posee un beneficio B_i con i = 1, 2, …, n; este beneficio representa la cantidad de aristas que posee este vértice.

```
C = {}      //Conjunto de vértices donde se ponen cámaras
Mientras (G no sea vacío) hacer
      Calcular B para cada Vi
      Ordenar V según Bi
      C = C U {Vi}      //Agregar el primer vértice a C
      Eliminar las aristas de V1
Devolver C
```

La idea general de este algoritmo es calcular la cantidad calcular la cantidad de aristas que posee cada vértice, luego ordenarlos desde aquel vértice que tenga mayor cantidad de aristas hasta el que tenga menos. Luego se pone una cámara en el primer vértice de la lista de vértices y se eliminan las aristas que este vértice poseía, representando que las calles colindantes a esta esquina ya están cubiertas. Este proceso de calcular, ordenar, agregar y eliminar se repetirá hasta que se hayan cubierto todas calles, en este caso sería hasta eliminar todas las aristas del grafo (poseer un grafo vacío). Esto se logra utilizando el método voraz como referencia.

b) El cálculo de complejidad siguiente, se estimo asumiendo que las aristas están representadas como una matriz de adyacencia:

- Comprobar que el grafo no sea vacío posee un $O(n^2)$, ya que en el peor de los casos se recorrerá completamente la matriz.
- Calcular B para cada V_i posee un $O(n^2)$, ya que es necesario recorrer toda la matriz.
- Ordenar V según B_i depende del algoritmo de ordenamiento utilizado, pensado en que se quiere disminuir el orden de complejidad, se considera merge sort para el ordenamiento que posee un $O(n * log(n))$ en el peor de los casos.
- Para eliminar las aristas de V_i solo es necesario recorrer la fila del vértice y cuando se elimine una arista en la posición (i, j), también se eliminará en la posición (j, i), por lo que posee un $O(n)$.

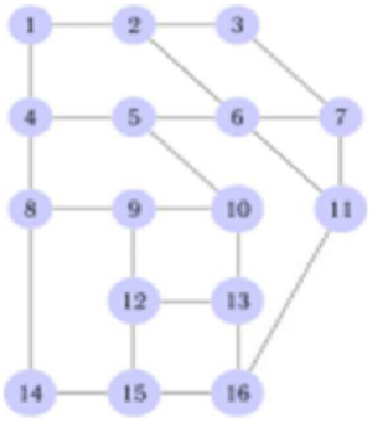
Teniendo en cuenta que todo lo anterior se encuentra dentro de un ciclo que en el peor de los casos se ejecutará $n/2$ veces (grafo con vértices conectados solo en pares o en una línea uno tras de otro), posemos decir que el algoritmo planteado posee el siguiente orden de complejidad: $O(n^3)$.

Si bien el algoritmo propuesto soluciona el problema con un orden no complejidad no tan elevado, este quizás podría ser mejorado cambiando la manera en la que se representan las aristas, ya que los algoritmos que operan sobre estas son los que poseen un orden de complejidad más elevado.

Nombre: Christofer Rodríguez Rut: 20.239.786-7 Sección: B-2

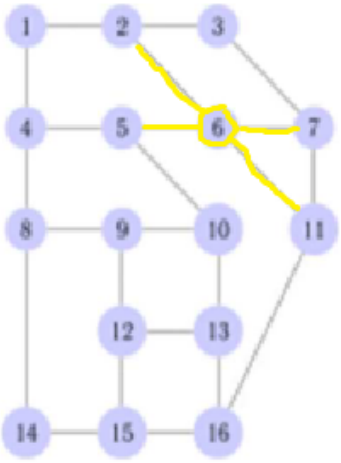
Pequeña traza

1)



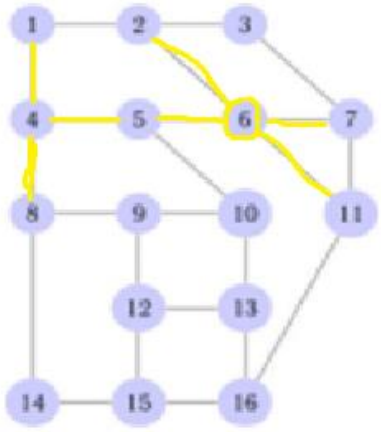
V6=4A
V2=3A
V4=3A
V5=3A
V7=3A
V8=3A
V9=3A
V10=3A
V11=3A
V12=3A
V13=3A
V15=3A
V16=3A
V1=2A
V3=2A
V14=2A

2)



V4=3A
V8=3A
V9=3A
V10=3A
V12=3A
V13=3A
V15=3A
V16=3A
V1=2A
V2=2A
V3=2A
V5=2A
V7=2A
V11=2A
V14=2A

3)



V10=3A
V12=3A
V13=3A
V15=3A
V16=3A
V2=2A
V3=2A
V7=2A
V8=2A
V11=2A
V14=2A
V1=1A
V5=1A