Informe TP1 - Teoría de Algoritmos

$Cuatrimestre$	1
$Año$	2022
$Estudiante$	Aldana Rastrelli
$Padrón$	98.408
$Fecha\ de\ Entrega$	6 de Abril de 2022
$N^{\circ}\ de\ entrega$	2da

▼ Índice

Enunciado

Parte 1: La señal para la ruta

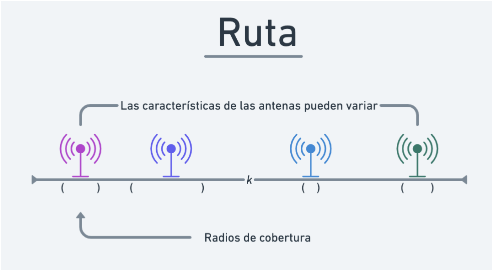
Para una ruta recién inaugurada de “k” kilómetros de longitud se debe construir un conjunto de antenas celulares para cubrir su recorrido. Se recibieron un conjunto de “n” propuestas. Cada propuesta corresponde a la instalación de una antena en una ubicación determinada. Las características de las antenas pueden variar. Sabemos por la información de cada contrato dónde se ubicará la antena y la cantidad de kilómetros que cubrirá de la ruta expresado en un radio de una cantidad de kilómetros desde donde está ubicada.

Nos solicitan seleccionar el menor subconjunto de contratos de forma que toda la ruta quede totalmente cubierta o que informe que esto no es posible.

Resuelva:

1. Proponga al menos 2 estrategias greedy que no sean óptimas para resolver el problema. Ejemplifique. ¿Por qué considera que son del tipo greedy?
2. Proponga una estrategia greedy que sea óptima. Justifique su optimalidad.
3. Explique cómo implementar algorítmicamente esa estrategia. Brinde pseudocódigo y estructuras de datos a utilizar.
4. Analice complejidad temporal y espacial de su propuesta
5. Programe su algoritmo.
6. Analice el resultado que obtiene mediante su algoritmo. ¿Puede encontrar alguna característica que beneficie algunos contratos frente a otros? ¿Existe alguna otra solución óptima alternativa?
7. Su programa mantiene la complejidad espacial y temporal de su algoritmo?

Resolución



Vamos a tener n propuestas diferentes de antenas, para cubrir k kilómetros de ruta.

1.

💡 Proponga al menos 2 estrategias greedy que no sean óptimas para resolver el problema. Ejemplifique. ¿Por qué considera que son del tipo greedy?

Para ejemplificar las soluciones `no` óptimas, se usará el ejemplo del enunciado “contratos.txt”. Entonces, antes de avanzar con las propuestas que `no` son óptimas, se busca una solución óptima de “contratos.txt” suponiendo $k = 552$ para poder comparar con ella. Esto se explicará luego (spoiler alert: tiene que ver con Dijsktra) pero la solución óptima que se encontró es:

```
5,80,80
6,300,160
2,402,150
```

Opción 1 no óptima

Se ordena de forma estable **según posición de ubicación en la ruta, de forma ascendente**.

Desde la primer antena, se eligen las subsiguientes a partir de la siguiente heurística:

> *Le seguirá aquella que se superponga lo menos posible con la actual, condicionando que la superposición debe ser por una cantidad de kilómetros mayor o igual a cero -es decir, no pueden quedar kilómetros sin cubrir entre antenas-.*

Esto se hace calculando el kilómetro a partir del cual la antena comienza con su cobertura (s_i) y el kilómetro a partir del cual finaliza la misma (t_i), siendo:

$$s_i = \begin{cases} 0 & \text{; si } ubicación - radio < 0 \\ ubicación - radio < 0 & \text{; en otro caso} \end{cases}$$
$$t_i = ubicacion + radio$$

Se toma como ejemplo el “contratos.txt” del enunciado, y se calcula el s_i y t_i para cada una de las antenas:

```
km, ubicacion, radio => s_i, t_i
1,44,50 => 0, 94
2,402,150 => 252, 552
3,219,35 => 184, 254
4,100,80 => 20, 180
5,80,80 => 0, 160
6,300,160 => 140, 460
7,150,30 => 120, 180
```

Ordenando, se obtiene:

```
1,44,50 => 0, 94
5,80,80 => 0, 160
4,100,80 => 20, 180
7,150,30 => 120, 180
3,219,35 => 184, 254
6,300,160 => 140, 460
2,402,150 => 252, 552
```

Mientras que el resultado final de antenas sería:

```
1,44,50 => 0, 94
4,100,80 => 20, 180
6,300,160 => 140, 460
2,402,150 => 252, 552
```

Se considera que es del tipo $greedy$ porque se busca siempre, para la siguiente antena, la mejor opción según la heurística ya mencionada.
Esta opción no es óptima ya que depende del orden en el que se ubicaron las antenas en el archivo inicial (el ordenamiento es estable) y no compara antenas que comienzan en la misma ubicación. Por ejemplo, hubiese convenido para utilizar menor cantidad de antenas, comenzar con la opción 5, pero por el orden dado se comenzó con la opción 1 y directamente se buscó aquella que superponga menos con esta.

Opción 2 no óptima: Mochila Fraccionaria

Ordenando la lista de opciones **según el radio de forma descendente** y de forma estable, se colocarán las antenas una a una en la ruta, desde aquella que tiene el radio más grande de todos hasta la que tiene el radio más pequeño. El punto de corte de la iteración será cuando la ruta ya no tenga kilómetros sin cobertura.
Se supone que $k = 552$.

Siguiendo con el ejemplo del enunciado, ordenándolo según el criterio:

```
6,300,160 => 140, 460
2,402,150 => 252, 552
4,100,80 => 20, 180
5,80,80 => 0, 160
1,44,50 => 0, 94
3,219,35 => 184, 254
7,150,30 => 120, 180
```

Entonces queda:

Iteración	Opción Elegida	s_i, t_i	Intervalos no cubiertos	Intervalos cubiertos
0	6,300,160	140,460	(0,140),(460, 552)	[140,460]
1	2,402,150	252, 552	(0,140)	[140,552]
2	4,100,80	20,180	(0, 20)	[20, 552]
3	5,80,80	0, 160	-	[0, 552]

- Sólo se conectan los vértices que están superpuestos con una cantidad de kilómetros mayor o igual a cero. Si los vértices no se superponen, los mismos no se conectan para no obtener pesos negativos.

Nuevamente se ve que la superposición entre las antenas es bastante grande. Y otra vez, como ya se conoce que hay una mejor solución de tres elementos, podemos afirmar que al menos esta no es la solución óptima.

Se considera que es greedy porque se busca, siempre y cuando la ruta tenga kilómetros sin cubrir, elegir siempre para la siguiente antena la mejor opción con la siguiente heurística:

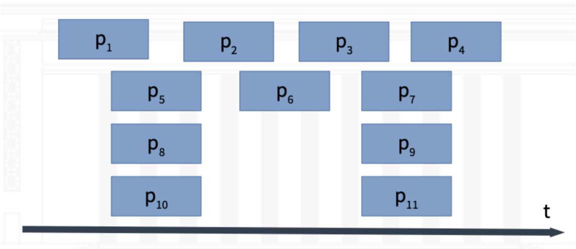
> *Aquella que tenga mayor radio de cobertura. Así, la idea es utilizar la menor cantidad de antenas cubriendo la mayor cantidad de kilómetros posibles.*

Se nota que no es óptima porque la ubicación de las antenas ya viene dada y quedan kilómetros sin cubrir que deben seguir llenándose con antenas. Si se posicionaran de forma conntinuada según este órden en el que las vamos eligiendo y con una superposición de 0kms, sería óptimo y similar al ejemplo de las patrullas de policía hecho en clase.

2.

💡 Proponga una estrategia $greedy$ que sea óptima. Justifique su optimalidad.

La primera idea que surgió es que el ejercicio podría llegar a realizarse con una suerte de $Interval\ Scheduling$, sólo que en vez de utilizar tiempo, utilizamos kilómetros; y en vez de buscar aquellas antenas que contengan un radio lo más pequeño posible, buscar aquellas con el radio más grande y tratar de **que se superpongan lo menos posible** -a diferencia de lo que se busca en Interval Scheduling, que es que directamente no se superpongan-.



Usando esa idea original como base y además tomando en cuenta una de las ideas no óptimas previas como inspiración, se procede con la siguiente forma, descompuesta en pasos. Considerando:

- El kilómetro a partir del cual la antena comienza con su cobertura (s_i)
- El kilómetro a partir del cual finaliza la misma (t_i)

Siendo ambos los definidos previamente en la opción no óptima n° 1.

Se siguen los siguientes pasos:

1. Ordenar según la posición en la que inicia la cobertura s_i de forma ascendente.
   El ordenamiento debe ser de forma estable.
2. Comenzar con la antena que tenga el s_i más chico. Si hubiese empate, se elige a aquella antena con el radio más grande (o análogo, el t_i más grande), cubriendo más superficie.
3. Iteraciones:
   a. Recorrer las opciones posteriores con la finalidad de encontrar aquella cuyo s_i' se superponga lo menos posible con el t_i de la antena inicialmente seleccionada.
   b. **La superposición debe tener un valor mayor o igual a cero**, tal que las antenas no dejen espacios entre sí sin cobertura.
   c. **Criterio de selección**: si el s_i' de la antena iterada es mayor al t_i de la antena inicialmente seleccionada -es decir, sus radios de cobertura dejan espacios entre sí-, se decide utilizar la antena anterior con la cual se superponía de forma mínima.

💡 Entonces la $heurística$ es:

> **Elegir siempre la antena que cumple con al menos uno de estos requisitos: O termina de cubrir la totalidad de los kilómetros (su t_i es mayor a k), o es la antena con mayor cobertura y menor superposición con la anteriormente elegida.**

Se analiza con el ejemplo, recordando que se supuso $k = 552$. Se ordena según el criterio:

```
1,44,50 => 0, 94
5,80,80 => 0, 160
4,100,80 => 20, 180
7,150,30 => 120, 180
6,300,160 => 140, 460
3,219,35 => 184, 254
2,402,150 => 252, 552
```

Luego, se itera y se eligen las antenas con menor superposición, obteniendo:

```
5,80,80 => 0, 160
6,300,160 => 140, 460
2,402,150 => 252, 552
```

Optimalidad

Aparenta ser imposible encontrar un contraejemplo así que se intentará demostrar la heurística:

No se puede asegurar que el conjunto mínimo de antenas A sea óptimo O porque podrían existir varias selecciones del mismo tamaño y de diferentes antenas que cubran la misma cantidad de kilómetros.

Lo que se quiere ver es que $|A| = |O|$, entonces se utilizará la idea de que el algoritmo greedy se mantiene “por delante” de la solución óptima.

Se pueden enumerar las antenas en el orden de selección $\{a_1, a_2, ..., a_k\}\ con\ |A| = k$, y asimismo los elementos de O: $\{o_1, o_2, ..., o_k\}\ con\ |O| = m$.

Sea t el km de finalización de cobertura de una antena y s el km donde la cobertura comienza:

Se quiere demostrar que para todo $r \leq k$ se cumple que $t(a_r) \geq t(o_k)$ (es decir, tiene mayor radio de cobertura).

Asimismo se tiene que $t(a_{r-1}) \geq t(o_{r-1})$, es decir, a_{r-1} tiene mayor radio de cobertura que o_{r-1}.

Como O está compuesto por antenas compatibles: $t(o_{r-1}) \geq s(o_r)$, es decir, se superponen.

Entonces, de estas dos ecuaciones, se obtiene que: $t(a_{r-1}) \geq s(o_r)$.

Es decir, en el momento en el que se seleccionó a_r, también estaba disponible o_r.

Greedy selecciona a aquel disponible con mayor radio de cobertura.

Entonces: $t(a_r) \geq t(o_r)$. Por lo tanto, se puede afirmar que el algoritmo greedy retorna un set óptimo A.

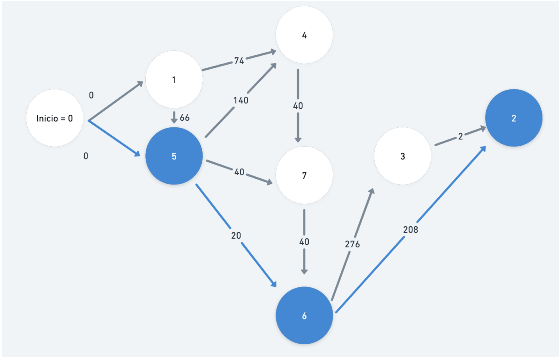
3.

💡 Explique cómo implementar algorítmicamente esa estrategia. Brinde pseudocódigo y estructuras de datos a utilizar.

Este método, cuando los kilómetros de la ruta son iguales al límite de cobertura de la antena más alejada, es análogo a **calcular el camino mínimo desde la antena más cercana al kilómetro 0 hasta la antena más alejada del origen**, considerando que:

- Los pesos entre vértices es la cantidad de kilómetros en los cuáles se superponen **(mayor o igual a cero, debe** existir una superposición. De no haber superposición, no existe el vértice).

- Se busca con el algoritmo de Dijkstra el camino mínimo desde cualquier vértice con inicio en 0 hasta el vértice con ubicación más grande (considerando que este siempre debe tener un $s_i(inicio\ de\ cobertura) < k$ y un $t_i(final\ de\ cobertura) \geq k$).



Cuando la ruta es más corta que la cantidad de antenas, se procede de la misma forma descartando aquellas antenas que no son relevantes para la ruta (su $s_i \geq k$).

> ❗ Si fuesemos a implementar el algoritmo Dijsktra de forma directa (lo cual no se hace en el código), encontraríamos siempre la solución óptima porque:
>
> - El grafo análogo al problema es dirigido y ponderado.
> - El objetivo es encontrar el camino con el menor peso total.
> - El gráfico no contiene ciclos negativos.
>
> Por lo tanto y dadas estas condiciones, Dijkstra siempre encontraría el mejor camino.

Pseudocódigo

```
Sea A el set de antenas:
    Ordenar A por su posición de inicio de cobertura s_i.

Sea AntenasElegidas el set de Antenas Elegidas.

Sea k los kilómetros de la ruta.

"anterior" = primer antena

Mientras que haya antenas por evaluar:
          Comparar las primeras dos antenas.
          Si empiezan en el mismo lugar, nos quedamos con la que tiene más cobertura y se guarda en "anterior"
              Continuar iteración

      Agregar "anterior" a AntenasElegidas.
      Guardar "anterior" en "ultima_elegida".

      ## Si la antena actual ya no está dentro de la ruta, las siguientes tampoco lo estarán (están ordenadas).
      Si la antena "actual" comienza su cobertura más alla de k:
              terminar las iteraciones.

      ## Si la última antena elegida y la actual no se superponen, entonces usar la antena anterior (que sí se superpone).
      Si "actual" y "ultima_elegida" no se superponen:
            Guardar "anterior" en AntenasElegidas.
            la "ultima_elegida" pasa a ser la "anterior"

      ## Si no quedan espacios sin cobertura, seguimos evaluando para ver si encontramos una mejor opción
      De lo contrario:
            Se analiza la antena siguiente avanzando:
            "anterior" = "actual"
            "actual" = siguiente antena de "actual"

Si terminaron las iteraciones y la ruta está incompleta, guardar "anterior" en AntenasElegidas.
```

Para recorrer las antenas ordenadas, se usa una cola por dos razones:

1. La facilidad de obtener la siguiente antena, ya que se pueden ordenar previamente.

2. Que como sólo se necesitan guardar 3 datos (**última antena seleccionada, antena anterior evaluada y antena actual de la iteración**), se puede hacer con variables auxiliares y no hay necesidad de una estructura de datos más compleja.

4.

> 💡 Analice complejidad temporal y espacial de su propuesta.

Si se usa un ordenamiento como Merge Sort, el mismo implica una complejidad temporal de $O(nlogn)$ con una complejidad espacial de $O(n)$.

Por otro lado, la iteración es $O(n)$ porque en el peor de los casos se recorren todas las opciones de antenas una sola vez, por lo que la complejidad temporal del algoritmo óptimo termina siendo $O(nlogn)$, mientras que la complejidad espacial es $O(n)$.

5.

> 💡 Programe su algoritmo.

Se adjunta el código del programa.
Instrucciones para correrlo:

```
>> python main.py "nombre_archivo.txt" km
```

- km debe ser un número
- El archivo debe estar guardado dentro de la carpeta "Código".

6.

> 💡 Analice el resultado que obtiene mediante su algoritmo. ¿Puede encontrar alguna característica que beneficie algunos contratos frente a otros? ¿Existe alguna otra solución óptima alternativa?

El ordenamiento original de las antenas **no condiciona** el resultado del algoritmo, eso ya de por sí es algo favorable del algoritmo, dado que sería la característica más común que podría llegar a beneficiar a unos contratos por sobre otros.

No van a existir soluciones óptimas alternativas, ya que por la heurística explicada en el punto 2, si la antena actual cubre los kms necesarios, se elige esa. Si no, se busca que tenga la menor superposición con la anterior.

Por ejemplo, para $k = 90$ va a elegir la antena `1,44,50` $(s_i = 0, t_i = 94)$ mientras que para $94 < k \leq 160$ va a elegir la antena `5, 80, 80` $(s_i = 0, t_i = 160)$.

7.

> 💡 Su programa mantiene la complejidad espacial y temporal de su algoritmo?

El pograma mantiene en efecto la complejidad espacial y temporal, ya que primero se realizó el análisis y el pseudocódigo y luego en base a eso se programó la solución.