



Universidad de los Andes
Ingeniería de Sistemas y Computación
Estructuras de Datos y Algoritmos

Reto 4: Navegando por la Red

Documento de análisis de los requerimientos propuestos en el Reto 4
del curso

María Paula Alméciga Moreno
m.almeciga@uniandes.edu.co
202023369

Andrés Felipe Vargas Cuadros
af.vargasc1@uniandes.edu.co
202013817

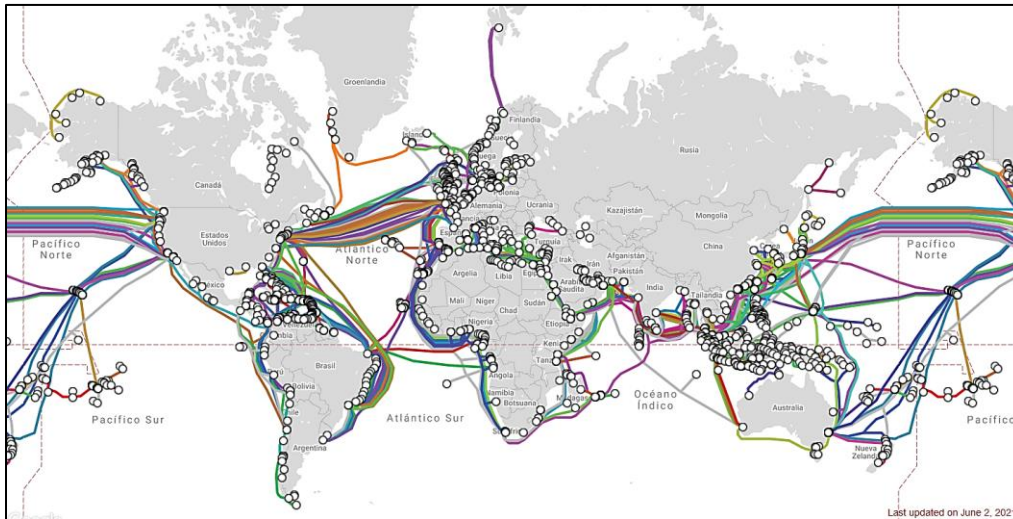
Bogotá, Colombia
Junio 2021

Contenido

Introducción	2
Requerimiento 1	3
Análisis de complejidad:	3
Pruebas:	3
Requerimiento 2	4
Análisis de complejidad:	4
Pruebas:	4
Requerimiento 3	6
Análisis de complejidad:	6
Pruebas:	6
Requerimiento 4	8
Análisis de complejidad:	8
Pruebas:	8
Requerimiento 5	9
Análisis de complejidad:	9
Pruebas:	9

Introducción

En el cuarto reto del curso, **Navegando por la Red**, se presenta una serie de datos con información sobre la red de cables submarinos que constituye la espina dorsal del Internet, al transportar el 99% de las telecomunicaciones digitales en el mundo, y posibilitar comunicación internacional a alta velocidad, con gran confiabilidad, y garantizando el funcionamiento mundial de la banda ancha.



Tomado de <https://www.submarinecablemap.com/>

Para el presente reto se consideran los siguientes objetivos:

- Utilizar grafos, en conjunto con las demás estructuras de datos del curso para solucionar los requerimientos del reto.
- Utilizar adecuadamente el patrón Modelo, Vista, Controlador.
- Utilizar adecuadamente el ambiente de trabajo (IDE, GIT y GitHub).

Para la implementación del reto se han propuesto los siguientes requerimientos:

1. Identificar los clústeres de comunicación.
2. Identificar los puntos de conexión críticos de la red.
3. La ruta de menor distancia.
4. Identificar la Infraestructura crítica de la red.
5. Análisis de fallas.

A continuación, se presenta información más detallada sobre cada uno de los requerimientos y un análisis de su rendimiento en términos de tiempo de ejecución y consumo de memoria, y análisis de complejidad.

Requerimiento 1

Se desea encontrar la cantidad de clústeres (componentes conectados) dentro de la red de cables submarinos y si dos landing points pertenecen o no al mismo clúster.

Entradas:

- Nombre del landing point 1.
- Nombre del landing point 2.

Salidas:

- Número total de clústeres en la red.
- Informar si dos landing points están en el mismo clúster o no.

Análisis de complejidad:

Se determinó que el análisis de complejidad en notación O es:

$$O(\quad)$$

Donde :

Pruebas:

A continuación, se presentan los resultados de la ejecución de una prueba específica. Adicionalmente, se incluyen los resultados de consumo de memoria y tiempo en la ejecución.

Requerimiento 2

Se desea encontrar el (los) landing point(s) que sirven como punto de interconexión a más cables en la red.

Entradas:

- Ninguna.

Salidas:

- Lista de landing points (nombre, país, identificador).
- Total de cables conectados a dichos landing points.

Análisis de complejidad:

Se determinó que el análisis de complejidad en notación O es:

$$O(n)$$

Donde n equivale a la cantidad de landing points existente, al ser recorridos en el siguiente ciclo del model:

```
points = mp.valueSet(catalog['landing_points'])
points_list = lt.newList()
for point in lt.iterator(points):
    amount = gr.degree(catalog['connections'], point['landing_point_id'])
    if len(point['name'].split(",")) > 1:
        element = {'name': str(point['name']), 'country': str((point['name'].split(",")[1])), 'id': point['landing_point_id'], 'amount':
    else:
        element = {'name': str(point['name']), 'country': str(point['name']), 'id': point['landing_point_id'], 'amount': amount}
    lt.addLast(points_list, element)
```

Fragmento de código en el model donde se evidencia la mayor complejidad

Pruebas:

A continuación, se presentan los resultados de la ejecución de una prueba específica. Adicionalmente, se incluyen los resultados de consumo de memoria y tiempo en la ejecución.

```
> Nombre: Akulivik, QC, Canada || País: QC || ID: 15046 || Cables: 1
> Nombre: Plerin, France || País: France || ID: 5836 || Cables: 2
> Nombre: San Sebastian de la Gomera, Canary Islands, Spain || País: Canary Islands || ID: 17558 || Cables: 3
> Nombre: Bandar Bukit Tinggi, Malaysia || País: Malaysia || ID: 14269 || Cables: 1
> Nombre: Sangata, Indonesia || País: Indonesia || ID: 14609 || Cables: 2
> Nombre: Sungsang, Indonesia || País: Indonesia || ID: 17288 || Cables: 1
> Nombre: Masohi, Indonesia || País: Indonesia || ID: 16625 || Cables: 2
> Nombre: Güimar, Canary Islands, Spain || País: Canary Islands || ID: 16017 || Cables: 2
> Nombre: St. Croix, VI, United States || País: VI || ID: 5750 || Cables: 9
> Nombre: Seward, AK, United States || País: AK || ID: 5755 || Cables: 2
> Nombre: Schoelcher, Martinique || País: Martinique || ID: 17024 || Cables: 1
> Nombre: Cabinda, Angola || País: Angola || ID: 10857 || Cables: 1
> Nombre: Chongming, China || País: China || ID: 5802 || Cables: 8
> Nombre: Juneau, AK, United States || País: AK || ID: 4816 || Cables: 1
> Nombre: Clarendville, NL, Canada || País: NL || ID: 10399 || Cables: 1
> Nombre: Kihei, Maui, HI, United States || País: Maui || ID: 7795 || Cables: 3
> Nombre: Sandy Point, Bahamas || País: Bahamas || ID: 5788 || Cables: 3
> Nombre: Nakhodka, Russia || País: Russia || ID: 5362 || Cables: 1
> Nombre: Chania, Greece || País: Greece || ID: 5723 || Cables: 8
> Nombre: Brighton, United Kingdom || País: United Kingdom || ID: 3170 || Cables: 1
> Nombre: Tiakur, Indonesia || País: Indonesia || ID: 16683 || Cables: 2
```

Algunos de los resultados impresos en consola

```
> Tiempo y memoria consumidos:
> Tiempo [ms]: 40.483 || Memoria [kB]: 677.737
```

Tiempo y memoria consumidos en la ejecución

Requerimiento 3

Se desea encontrar la ruta mínima en distancia para enviar información entre dos países, los puntos de origen y destino serán los landing point de la ciudad capital.

Entradas:

- Nombre del País A.
- Nombre del País B.

Salidas:

- Ruta (incluir la distancia de conexión [km] entre cada par consecutivo de landing points).
- Distancia total de la ruta.

Análisis de complejidad:

Se determinó que el análisis de complejidad en notación O es:

$$O(n)$$

Pues es la mayor complejidad de la implementación (siendo n la cantidad de landing points existentes), pues si bien en la resolución principal del problema se obtiene una complejidad de $O(E \log V)$ con la ejecución del algoritmo de Dijkstra sobre el grafo de conexiones, antes de ello se usa el siguiente ciclo en el controller para determinar un Landing Point para cada país dado:

```
for lp in lt.iterator(catalog['lpoints_list']):
    if countryA in lp and conta == True:
        pointA = (lp.split(";"))[0]
        conta = False
    elif countryB in lp and contB == True:
        pointB = (lp.split(";"))[0]
        contB = False
    if (conta == False and contB == False) or (count == int(lt.size(catalog['lpoints_list'])) - 1):
        break
    count += 1
```

Fragmento de código en el controller donde se evidencia la mayor complejidad

Pruebas:

A continuación, se presentan los resultados de la ejecución de una prueba específica. Adicionalmente, se incluyen los resultados de consumo de memoria y tiempo en la ejecución.

```
===== REQUERIMIENTO 3 =====  
Ingrese el país A: Nigeria  
Ingrese el país B: Egypt  
  
La ruta obtenida (por tramos) ha sido:  
> 3280 a 3117 con una distancia de 4256.16km  
> 3117 a 4177 con una distancia de 4892.47km  
> 4177 a 4361 con una distancia de 1177.55km  
> 4361 a 5950 con una distancia de 291.22km  
> 5950 a 9487 con una distancia de 1385.8km  
La distancia total entre ambos países es de 12003.2km  
  
-----  
> Tiempo y memoria consumidos:  
> Tiempo [ms]: 307.233 || Memoria [kB]: 48.215
```

Resultado obtenido en consola, con tiempo y memoria consumidos en la ejecución

Requerimiento 4

Se desea encontrar la infraestructura crítica para poder garantizar el mantenimiento preventivo del mismo. Para tal fin se requiere que se identifique la red de expansión mínima en cuanto a distancia que pueda darle cobertura a la mayor cantidad de landing point de la red.

Entradas:

- Ninguna.

Salidas:

- El número de nodos conectados a la red de expansión mínima.
- El costo total (distancia en [km]) de la red de expansión mínima.
- Enunciar la conexión más larga (distancia en [km]) que hace parte de la red de expansión mínima.
- Enunciar la conexión más corta (distancia en [km]) que hace parte de la red de expansión mínima.

Análisis de complejidad:

Se determinó que el análisis de complejidad en notación O es:

$$O(\quad)$$

Donde :

Pruebas:

A continuación, se presentan los resultados de la ejecución de una prueba específica. Adicionalmente, se incluyen los resultados de consumo de memoria y tiempo en la ejecución.

Requerimiento 5

Se desea conocer el impacto que tendría el fallo de un determinado landing point que afecta todos los cables conectados al mismo. Para tal fin se requiere conocer la lista de países que podrían verse afectados al producirse una caída en el proceso de comunicación con dicho landing point.

Entradas:

- Nombre del landing point.

Salidas:

- Número total de países afectados.
- Lista de los países afectados (la lista debe darse por distancia en km descendentemente)

Análisis de complejidad:

Se determinó que el análisis de complejidad en notación O es:

$$O(n)$$

Donde n equivale a la cantidad de landing points existente, al ser recorridos en el siguiente ciclo del model:

```
for adjacent in lt.iterator(points):
    distance = (gr.getEdge(catalog['connections'], lpoint, adjacent))['weight']
    name = ((mp.get(catalog['landing_points'], adjacent))['value']['name'].split(",")[1]
    if not lt.isPresent(points_list, [name, distance]):
        lt.addLast(points_list, [name, distance])
```

Fragmento de código en el model donde se evidencia la mayor complejidad

Pruebas:

A continuación, se presentan los resultados de la ejecución de una prueba específica. Adicionalmente, se incluyen los resultados de consumo de memoria y tiempo en la ejecución.

```
===== REQUERIMIENTO 5 =====  
Ingrese el nombre del landing point: Barcelona, Spain  
  
La cantidad de países afectados directamente sería: 2  
> United Kingdom a 1169.41km de distancia  
> Ghana a 3990.77km de distancia  
  
-----  
> Tiempo y memoria consumidos:  
> Tiempo [ms]: 0.356 || Memoria [kB]: 8.397
```

Resultado obtenido en consola, con tiempo y memoria consumidos en la ejecución