

Trabajo Práctico 2

[7529/9506] Teoría de Algoritmos I
Segundo cuatrimestre de 2021

Grupo:	404
Repositorio:	github.com/lucashemmingsen/7529tp2
Entrega:	nº 1 (20/10/2021)

Integrantes del grupo 404

Padrón	Apellido, Nombre	Email
76187	Xxxxxxxxxx, Xxxxx Xxxxx	xxxxxxxxxxxx@fi.uba.ar
97529	Xxxxxxxx, Xxxxx Xxxxxxx	xxxxxxxxxx@fi.uba.ar
102593	Xxxxxx, Xxxxx Xxxxx	xxxxxxx@fi.uba.ar
102649	Xxxxxx, Xxxxxxx Xxxxx	xxxxxxx@fi.uba.ar
106713	Xxxx Xxxxx, Xxxxx	xxxxx@fi.uba.ar

Índice

1. Introducción	2
1.1. Resumen	2
1.2. Lineamientos básicos	2
2. Parte 1: Minimizando costos	3
2.1. Enunciado	3
2.1.1. Formato de los archivos	3
2.2. Algoritmo de Johnson: funcionamiento	4
2.3. Comparación de algoritmos: Ventajas y desventajas	5
2.4. Comparación de algoritmos: Ventajas y desventajas	6
2.5. Algoritmo de Johnson: resolución manual	7
2.6. Metodología Greedy	8
2.7. Programación dinámica	9
2.8. Programa con solución	10
3. Parte 2: Un poco de teoría	11
3.1. Enunciado	11
3.2. Formas de resolución de problemas	12
3.3. Caso teórico puntual	13

1. Introducción

1.1. Resumen

El presente informe documenta el enunciado y la solución del segundo trabajo práctico de la materia Teoría de Algoritmos I. El mismo comprende el análisis del problema planteado, la comparación de posibles algoritmos y la resolución manual del algoritmo de Johnson; así como también respuestas a preguntas teóricas.

1.2. Lineamientos básicos

- El trabajo se realizará en grupos de cinco personas.
- Se debe entregar el informe en formato pdf y código fuente en (.zip) en el aula virtual de la materia.
- El lenguaje de implementación es libre. Recomendamos utilizar C, C++ o Python. Sin embargo si se desea utilizar algún otro, se debe pactar con los docentes.
- Incluir en el informe los requisitos y procedimientos para su compilación y ejecución. La ausencia de esta información no permite probar el trabajo y deberá ser re-entregado con esta información.
- El informe debe presentar carátula con el nombre del grupo, datos de los integrantes y y fecha de entrega. Debe incluir número de hoja en cada página.
- En caso de re-entrega, entregar un apartado con las correcciones mencionadas

2. Parte 1: Minimizando costos

2.1. Enunciado

Una empresa productora de tecnología está planeando construir una fábrica para un producto nuevo. Un aspecto clave en esa decisión corresponde a determinar dónde la ubicarán para minimizar los gastos de logística y distribución. Cuenta con N depósitos distribuidos en diferentes ciudades. En alguna de estas ciudades es donde deberá instalar la nueva fábrica. Para los transportes utilizarán las rutas semanales con las que ya cuentan. Cada ruta une dos depósitos en un sentido. No todos los depósitos tienen rutas que los conecten. Por otro lado, los costos de utilizar una ruta tienen diferentes valores. Por ejemplo hay rutas que requieren contratar más personal o comprar nuevos vehículos. En otros casos son rutas subvencionadas y utilizarlas les da una ganancia a la empresa. Otros factores que influyen son gastos de combustibles y peajes. Para simplificar se ha desarrollado una tabla donde se indica para cada ruta existente el costo de utilizarla (valor negativo si da ganancia).

Los han contratado para resolver este problema.

Han averiguado que se puede resolver el problema utilizando Bellman-Ford para cada par de nodos o Floyd-Warshall en forma general. Un amigo les sugiere utilizar el algoritmo de Johnson. Aclaración: ¡No existen ciclos negativos!

Se pide:

1. Investigar el algoritmo de Johnson y explicar cómo funciona. ¿Es óptimo?
2. En una tabla comparar la complejidad temporal y espacial de las tres propuestas.
3. Analizar en qué situaciones una solución es mejor que otras.
4. Crear un ejemplo con 5 depósitos y mostrar paso a paso cómo lo resolvería el algoritmo de Johnson.
5. ¿Puede decirse que Johnson utiliza en su funcionamiento una metodología greedy? Justifique.
6. ¿Puede decirse que Johnson utiliza en su funcionamiento una metodología de programación dinámica? Justifique.
7. Programar la solución usando el algoritmo de Johnson.

2.1.1. Formato de los archivos

Formato de los archivos: El programa debe recibir por parámetro el path del archivo donde se encuentran los costos entre cada depósito. El archivo debe ser de tipo texto y presentar por renglón, separados por coma un par de depósitos con su distancia.

Ejemplo: "depositos.txt"

```
A,B,54
A,D,-3
B,C,8
...
```

Debe resolver el problema y retornar por pantalla la solución. Debe mostrar por consola en qué ciudad colocar el depósito. Además imprimir en forma de matriz los costos mínimos entre cada uno de los depósitos.

2.2. Algoritmo de Johnson: funcionamiento

Enunciado 1.1

Investigar el algoritmo de Johnson y explicar cómo funciona. ¿Es óptimo?

2.3. Comparación de algoritmos: Ventajas y desventajas

Enunciado 1.2

En una tabla comparar la complejidad temporal y espacial de las tres propuestas.

2.4. Comparación de algoritmos: Ventajas y desventajas

Enunciado 1.3

Analizar en qué situaciones una solución es mejor que otras.

2.5. Algoritmo de Johnson: resolución manual

Enunciado 1.4

Crear un ejemplo con 5 depósitos y mostrar paso a paso cómo lo resolvería el algoritmo de Johnson.

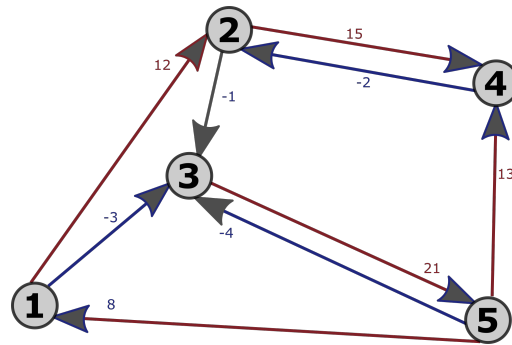


Figura 1: Grafo G (original).

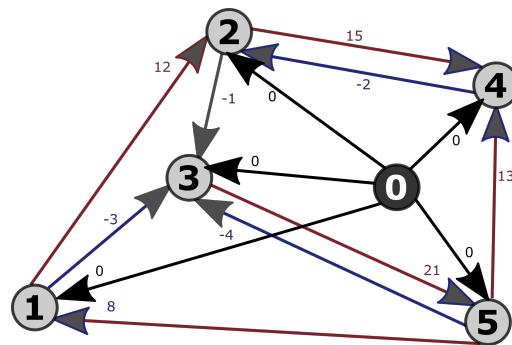


Figura 2: Grafo G' (con nodo temporal 0).

2.6. Metodología Greedy

Enunciado 1.5

¿Puede decirse que Johnson utiliza en su funcionamiento una metodología greedy? Justifique.

2.7. Programación dinámica

Enunciado 1.6

¿Puede decirse que Johnson utiliza en su funcionamiento una metodología de programación dinámica? Justifique.

2.8. Programa con solución

Enunciado 1.7

Programar la solución usando el algoritmo de Johnson.

3. Parte 2: Un poco de teoría

3.1. Enunciado

1. Hasta el momento hemos visto 3 formas distintas de resolver problemas. Greedy, división y conquista y programación dinámica.
 - a) Describa brevemente en qué consiste cada una de ellas.
 - b) Identifique similitudes, diferencias, ventajas y desventajas entre las mismas. ¿Podría elegir una técnica sobre las otras?
2. Tenemos un problema que puede ser resuelto por un algoritmo Greedy (G) y por un algoritmo de Programación Dinámica (PD). G consiste en realizar múltiples iteraciones sobre un mismo arreglo, mientras que PD utiliza la información del arreglo en diferentes subproblemas a la vez que requiere almacenar dicha información calculada en cada uno de ellos, reduciendo así su complejidad; de tal forma logra que $O(PD) < O(G)$. Sabemos que tenemos limitaciones en nuestros recursos computacionales (CPU y principalmente memoria). ¿Qué algoritmo elegiría para resolver el problema?

Pista: probablemente no haya una respuesta correcta para este problema, solo justificaciones correctas.

3.2. Fromas de resolución de problemas

Enunciado 2.1

Hasta el momento hemos visto 3 formas distintas de resolver problemas. Greedy, división y conquista y programación dinámica.

1. Describa brevemente en qué consiste cada una de ellas.
2. Identifique similitudes, diferencias, ventajas y desventajas entre las mismas. ¿Podría elegir una técnica sobre las otras?

3.3. Caso teórico puntual

Enunciado 2.2

Tenemos un problema que puede ser resuelto por un algoritmo Greedy (G) y por un algoritmo de Programación Dinámica (PD). G consiste en realizar múltiples iteraciones sobre un mismo arreglo, mientras que PD utiliza la información del arreglo en diferentes subproblemas a la vez que requiere almacenar dicha información calculada en cada uno de ellos, reduciendo así su complejidad; de tal forma logra que $O(PD) < O(G)$. Sabemos que tenemos limitaciones en nuestros recursos computacionales (CPU y principalmente memoria). ¿Qué algoritmo elegiría para resolver el problema?