



Condiciones

- Este proyecto es para ser realizado en grupos de máximo tres (3) estudiantes.
- La entrega del proyecto debe incluir
 - Un informe (documento PDF) en formato IEEE.
 - El código fuente de los algoritmos implementados
 - Los archivos con los datos utilizados para la ejecución y análisis experimental de los algoritmos.
- La entrega se debe hacer a través del Campus Virtual a través del enlace creado para dicho fin. En mismo enlace especifica la fecha de entrega del proyecto.

Introducción

Una empresa de mensajería local (solo hace entregas en la ciudad) necesita optimizar su proceso de entrega de encomiendas y para ello acudido a los estudiantes de una prestigiosa universidad para que implementen un producto de software que permita especificar el orden en que deben ser entregadas las encomiendas.

Para mayor simplicidad del problema se tienen las siguientes restricciones:

- Las entregas serán realizadas por un único mensajero. En caso de haber más mensajeros, lo que se hace es que la ciudad se subdivide en regiones, cada una de ellas asignada a un mensajero (no se trabajará para efectos del proyecto). Además, cada encomienda tiene un valor que representa el dinero que el mensajero recibe por su entrega.
- Se asumirá que la ciudad es una cuadrícula de cuadras, con calles y carreras, donde el tiempo para desplazarse entre calles y carreras será el mismo para toda la ciudad. A continuación la representación de una ciudad de 5x6 manzanas (en amarillo las cuadras donde se deben entregar las encomiendas y en azul la ubicación de la oficina).

	1	2	3	4	5
1					1
2					
3	2		6		
4					
5		3			
6	4				5

Si se ignoran las calles, la ciudad se puede ver como una matriz. Por ejemplo:

	1	2	3	4	5
1					1
2					
3	2		6		
4					
5		3			
6	4				5

- No se asumirán direcciones concretas (Calle X Cra Y). Para efectos de especificar la dirección de entrega de una encomienda, se usará un sistema coordinado el cual especifica la manzana en donde se debe realizar una entrega. Por ejemplo, al encomienda No 4 se debe entregar en la manzana (6,1); la entrega 1 se realiza en la manzana (1,5). La manzana en azul (ubicación de la oficina) corresponde a la manzana (2,4).
- A partir de la ubicación de las encomiendas, usted debe calcular la distancia entre la oficina y cada una de ellas, y entre ellas mismas. Para el ejemplo anterior, la matriz (simétrica) de distancias es:

	0	1	2	3	4	5	6
0	0	2	4	5	7	5	2
1	2	0	6	7	9	5	4
2	4	6	0	3	3	7	2
3	5	7	3	0	2	4	3
4	7	9	3	2	0	4	5
5	5	5	7	4	4	0	5
6	2	4	2	3	5	5	0

Problemas a Resolver

A continuación se listan los algoritmos que se deben diseñar :

- Realizar la entrega de todos los paquetes en el menor tiempo posible. Para este caso, no es importante el valor que recibe el mensajero por realizar las entregas.
- Realizar las entregas que le representen mayor beneficio al mensajero, en una cantidad de tiempo determinado (por ejemplo, diez unidades de tiempo).
- OPCIONAL, hacer uso de técnicas voraces para resolver el problema de entregar todas las encomiendas, sin restricción de tiempo.

Ingreso de Datos al Programa

El ingreso de los datos al programa se generará a partir de un programa que entregará el docente de la asignatura, el generará un archivo de texto plano (valores separados por coma) con la siguiente estructura:

- línea 1: tamaño de la ciudad (dos valores) y tiempo que se tarda en recorrer una calle/carrera
- línea 2: cantidad de encomiendas y tiempo del

mensajero

- línea 3: ubicación de la oficina
- línea 4 en adelante: ubicación de las encomiendas.

El archivo para el ejemplo descrito es:

```
6,5,1
6,10
2,4
1,5,3,1,5,2,6,1,6,5,3,3
```

El generador de casos de prueba lo puede descargar del Campus Virtual, en la pestaña “Evaluación”. Se trata de una archivo programa que debe ser ejecutado a través de una terminal de comandos (cmd para Windows y term o xterm para Linux)

El informe

El informe entregado debe contener para cada uno de los problemas (el grupo determinará el orden en que aparecerá estos items):

1. El algoritmo recursivo (pseudo-código) y las consideraciones o modificaciones importantes al momento de realizar la implementación.
2. La especificación de la estructura óptima
3. La ecuación de recurrencia para el calculo del costo óptimo
4. El algoritmo iterativo que calcula el costo óptimo (pseudo-código) y las consideraciones o modificaciones importantes al momento de realizar la implementación.
5. El algoritmo (pseudo-código) que reconstruye la solución óptima y las consideraciones o modificaciones importantes al momento de realizar la implementación.
6. El análisis teórico de los algoritmo diseñados
7. En análisis experimental de los algoritmos diseñados (incluyendo el gráfico con línea de tendencia y ecuación de la línea de tendencia)
8. Conclusiones