

Reparto a domicilio

Objetivos

- Diseñar un algoritmo basado en alguna de las estrategias de diseño de algoritmos vistas en el temario para resolver un problema de la forma más eficiente posible.
- Saber explicar el funcionamiento de un algoritmo de forma que sea entendible por una persona "ajena" a la asignatura.
- Saber escribir el pseudocódigo de un algoritmo.
- Implementar un algoritmo en C o C++ a partir de su pseudocódigo y verificar su correcto funcionamiento.
- Calcular la complejidad temporal asintótica de un algoritmo.
- Manejar la bibliografía de la materia, así como las diferentes fuentes y recursos on-line.

Enunciado

El empleado del servicio de reparto de un supermercado tiene que realizar el reparto de la compra realizada por diferentes clientes. Al empleado se le proporciona un listado con el listado de clientes que hay que visitar y cuánto se tarda en ir de un domicilio a otro. Se quiere implementar un programa que proporcione información sobre los recorridos que puede realizar el empleado para realizar el reparto saliendo y volviendo al supermercado, y proporcionando información sobre el tiempo de cada uno de ellos.

Actividades

Actividad 1: Memoria

Se realizará una memoria que contendrá:

- **PORTADA:**

Una hoja con el nombre de la asignatura, el nombre de la práctica ("Reparto a domicilio"), el curso académico (2021/2022), y los nombres y apellidos de los componentes del grupo.

- **ÍNDICE O TABLA DE CONTENIDO**

La siguiente página contendrá el índice del trabajo. En el índice se incluirán los apartados descritos a continuación **manteniendo los nombres y la numeración** y se incluirá el número de página donde comienzan.

1.- Algoritmo 1: obtener un listado de los posibles recorridos resaltando el más corto.

1.1.- Descripción y tipificación

Descripción detallada del funcionamiento de un algoritmo lo más eficiente posible que resuelva el problema y que se ajuste exclusivamente a lo que se pide en el enunciado. Esta descripción debe realizarse de forma que quede claro su funcionamiento y sea entendible por una persona ajena a la asignatura (por ejemplo, amistades que no estudien informática, hermanos/-as, padres,...).

Además se realizará la tipificación del problema (variables principales del problema, qué significan, restricciones, función objetivo).

1.2.- Estrategia de programación

Identificación de la/las estrategia/as general de programación que utiliza el algoritmo descrito en los apartados anterior y justificación.

1.3.- Pseudocódigo

Pseudocódigo del algoritmo utilizando la nomenclatura vista en la asignatura.

1.4. Ejemplo

Inclusión de un ejemplo de aplicación y traza detallada del algoritmo. El ejemplo será distinto al que se incluye en el enunciado de esta práctica. En ese apartado no se dará como válida una traza generada por el propio programa implementado y capturas de pantalla del mismo.

1.5.- Complejidad asintótica

Análisis y justificación de la complejidad asintótica del algoritmo del pseudocódigo.

2.- Algoritmo 2: obtener el camino más corto, mejorando el algoritmo 1.

Este algoritmo deberá obtener únicamente una solución al problema (no el listado completo) que deberá ser el camino más corto. Para ello, será obligatorio que este segundo algoritmo planteado sea una versión mejorada del algoritmo anterior (por ejemplo, generando menos soluciones hasta llegar a la óptima o eligiendo alguna propuesta que, aunque no se consiga el óptimo, se consiga una mejora significativa en cuanto a su complejidad).

A la hora de realizar la memoria de este segundo algoritmo, durante el desarrollo de los distintos apartados de la memoria se deben explicar las mejoras que proponéis con este algoritmo 2 respecto a la propuesta del algoritmo 1. Por tanto, se debe tener en cuenta que, si existen similitudes entre el Algoritmo 1 y el Algoritmo 2, en los distintos apartados se debe dedicar el esfuerzo a explicar las diferencias o aquellas partes que sean realmente nuevas; es decir, no se debe copiar y pegar literalmente aquellas partes desarrolladas en el punto 1 de la memoria y volver a repetirlas aquí sin aportar nada nuevo.

2.1.- Descripción y tipificación

2.2.- Estrategia de programación

2.3.- Pseudocódigo

2.4.- Ejemplo

2.5.- Complejidad asintótica

2.6.- Evidencias de las mejoras

Justificar si la mejora del algoritmo 2 frente al 1 es una mejora teórica y/o empírica y ofrecer alguna evidencia que la demuestre (se pueden usar las complejidades asintóticas obtenidas, incluir sentencias en el código que os permitan obtener estadísticas relevantes y mostrarlas, comparar tiempos de ejecución para diferentes tamaños del problema...).

3.- Bibliografía

Especificación de toda la bibliografía consultada con el siguiente formato:

- | | |
|-----------------------------|--|
| • Para libros, apuntes,...: | • Para direcciones web: |
| Título: | Título: |
| Autor/-es: | Url: (enlace directo a la página web) |
| Año: | |

El formato para la memoria es:

- Títulos de los apartados: Arial, 12
- Texto general: Arial, 10
- Interlineado: Sencillo o simple.
- Párrafo: Espaciado anterior y posterior: 0
- Márgenes: 2 cm. a cada lado.
- Las páginas irán numeradas y la numeración se corresponderá con la numeración de la tabla de contenido que se indica en el índice.

Actividad 2: Programa

Implementación de un programa en C/C++ correspondiente al pseudocódigo de los algoritmos de la memoria (siguiendo el modo de transformación visto en la asignatura). El programa presentará un menú con las siguientes opciones:

- 1.- Obtener lista de clientes.
- 2.- ¿Ocultar listado de recorridos en opción 3?
- 3.- Listado de recorridos resaltando el más corto.
- 4.- Recorrido más corto (algoritmo mejorado).
- 5.- Salir.

A continuación, se detalla qué debe realizar cada opción.

1.- Obtener lista de clientes.

Se pedirá el nombre de un fichero de texto sin formato (con extensión .txt) que contendrá los datos de la lista. La primera línea del fichero contendrá el número de clientes a los que hay que repartir. El resto de líneas tendrán el siguiente formato:

ORIGEN DESTINO TIEMPO donde:

- ORIGEN: es un número entero que identifica a un cliente o 0 para indicar que es el SUPERMERCADO.
- DESTINO: otro número entero para identificar al cliente o 0 para indicar SUPERMERCADO.
- TIEMPO: tiempo en minutos de ir de ORIGEN a DESTINO.

Los números de los clientes irán desde 1 hasta el número total de clientes, indicado en la primera línea del fichero.

Un posible ejemplo de fichero es:

```
3
0 1 20
1 2 50
2 3 50
0 2 40
0 3 60
3 0 20
1 0 30
1 3 25
2 0 20
2 1 25
3 1 30
```

2.- ¿Ocultar listado de recorridos en opción 3?

Se preguntará al usuario si al ejecutar la opción 3 se deben ocultar todos los recorridos encontrados mostrando únicamente el más corto. Si la opción de ocultar caminos está activada, al ejecutar la opción 3 se deberá mostrar en su lugar el número de caminos encontrados y la información del recorrido más corto. Por último, si se ejecuta la opción 3 sin haber pasado por dos, el comportamiento por defecto será mostrarlos.

3.- Listado de recorridos resaltando el más corto.

Se mostrará el listado con los diferentes recorridos resaltando el recorrido con menor tiempo de reparto. Para cada recorrido, se deberán indicar en orden los clientes que hay que visitar y el tiempo, según el formato que se indica a continuación. Para el fichero anterior hay 4 posibles recorridos:

Recorrido 1	Recorrido 2	Recorrido 3	Recorrido 4
SUPER	SUPER	SUPER	SUPER
1 20	2 40	2 40	3 60
2 50	1 25	3 50	1 30
3 50	3 25	1 30	2 50
SUPER 20	SUPER 20	SUPER 30	SUPER 20
Coste: 140	Coste: 110	Coste: 150	Coste: 160

Siendo el “Recorrido 2” el más corto y por tanto será éste el que se debe resaltar.

4.- Recorrido más corto (algoritmo mejorado).

Se deberá mostrar únicamente el recorrido más corto con el mismo formato que el apartado anterior. Adicionalmente, se deberán mostrar estadísticas (por ejemplo, el número total de recorridos que el programa ha tenido que calcular para encontrar la solución, el tiempo de ejecución empírico que tarda el programa en calcular dicho recorrido...) que servirán como evidencias a incluir en el apartado 2.6 de la memoria.

5.- Salir.

Con esta opción se saldrá del programa. Mientras no se pulse esta opción se volverá a presentar el menú del programa para que se pueda elegir cualquiera de las opciones del mismo.

Modo de entrega

La práctica se realizará **en grupos de 2 o 3 componentes**.

Se dispondrá de una tarea para su entrega en el campus virtual de la asignatura. Al menos uno de los componentes del grupo entregará un archivo comprimido llamado **proyectompa2122_evcontinua.zip**. Este archivo **deberá** contener los siguientes ficheros: reparto.pdf (memoria de la práctica), reparto.cpp (código fuente del programa) y reparto.exe (ejecutable). Se indicará el nombre de los miembros del grupo tanto en el.pdf como en el .cpp.

Fecha fin de entrega: 15 de junio a las 23:59.

Evaluación

Esta práctica constituye 2.0 puntos de la nota final de la asignatura. La distribución es la siguiente:

Algoritmo 1: listado de recorridos resaltando el más corto (1.4 puntos)

Algoritmo 2: recorrido más corto (algoritmo mejorado) (0.6 puntos)

La realización de un algoritmo comprende el desarrollo de la memoria y del programa.

Se valorará:

- Memoria
 - Desarrollo correcto de los apartados especificados.
 - Algoritmo propuesto, valorando que sea lo más eficiente posible.
 - Nivel de las mejoras logradas en el algoritmo 2.
 - Pseudocódigo según la notación vista de clase.
 - Claridad en la redacción, sin errores de sintaxis (faltas de ortografía, tildes,...).
- Programa
 - El correcto funcionamiento del programa.
 - El formato de salida por pantalla, claridad y facilidad de manejo del programa.
 - Adecuación de la implementación del algoritmo con el pseudocódigo de la memoria.
- Entrevista personal

Se podrá realizar una entrevista individual a los componentes del grupo donde el profesorado realizará preguntas sobre la práctica, de forma que las respuestas del alumno serán decisivas

para la nota final del alumno en la práctica, pudiendo haber diferentes puntuaciones entre los miembros de un grupo según esta entrevista.

IMPORTANTE:

La nota de la práctica será un 0 si:

- Faltan la mayoría de apartados de la memoria de la opción entregada.
- Se utiliza pseudocódigo que no se corresponde con el visto en la asignatura.
- Falta el programa que implemente el algoritmo/-s descritos en la memoria.
- El programa implementado no se corresponde con el pseudocódigo descrito en la memoria.
- El código del programa da errores de compilación utilizando la versión del compilador minGW para C/C++ instalada en el aula de prácticas.
- Se utilizan librerías no vistas en las prácticas de la asignatura sin solicitar y recibir su autorización por parte del profesorado.
- El programa no obtiene correctamente la solución del problema.
- Se detectan indicios de copia entre prácticas. En este caso la nota de **TODAS las prácticas implicadas** será un 0. No se contempla ningún tipo de justificaciones: “Hemos comentado cosas de la práctica entre nosotros,...”, “la práctica la he hecho yo y él/ella se la ha copiado”, “no sé cómo ha llegado la práctica a manos de otra persona”,....)
- El alumnado no defiende adecuadamente la práctica en la entrevista personal.

¡ ¡ ¡ COMPROBAD ESTOS PUNTOS ANTES DE ENTREGAR LA PRÁCTICA ! ! !