

Laboratorio 5

Todas las coordinaciones

FACULTAD DE INGENIERIA
Departamento de Ingeniería Informática
Análisis de algoritmos y estructuras de datos



24/05 /2017

Juan Estructuradedatos está cansado de tratar de resolver laberintos desde inicio de año (figura 1), ya que el señor Estructuradedatos no ha podido resolver todos los laberintos que ha solicitado una oscura organización llamada ADEsetnaduyA.

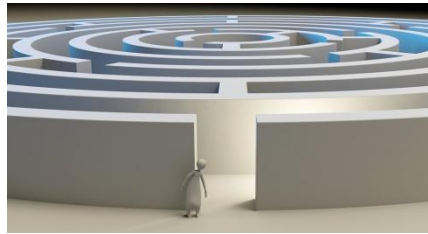


Figura 1: Juan Estructuradedatos y los Laberinto.

Afortunadamente el señor Estructuradedatos posee una rica red de contactos y la ha utilizado para poder resolver muchos problemas en el pasado, incluyendo los tediosos laberintos. Pero esta misma red de contactos, al ser tan basta, le complica al momento de tener que buscar el número de teléfono de alguien para poder solicitar ayuda. Es por esto que se ha contactado con usted, ya que se sabe que ha logrado superar los desafíos propuestos por los ADEsetnaduyA en anteriores oportunidades. Es así que Juan Estructuradedatos le solicita crear un programa que logre encontrar un número de teléfono a partir del nombre completo de una persona. Además, **le ha solicitado que utilice para este propósito un árbol AVL o un árbol 2-3**, ya que Estructuradedatos escucho que otra antigua leyenda de los ADEsetnaduyA dice que los AVL y 2-3 reducen el tiempo de búsqueda en comparación a estructuras como listas enlazadas. Sí, nosotros también creemos que esas leyendas son algo ñoñas para una organización, pero los ADEsetnaduyA nos caen bien, así que no le damos mucha importancia.

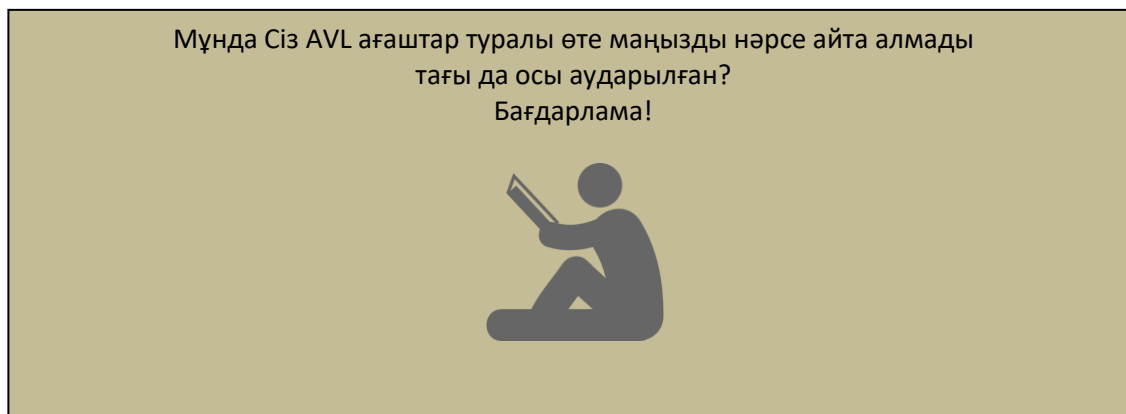


Figura 2: Juan Estructuradedatos y la leyenda.

Laboratorio 5

Todas las coordinaciones

FACULTAD DE INGENIERIA
Departamento de Ingeniería Informática
Análisis de algoritmos y estructuras de datos



24/05 /2017

Para esto, el programa debe recibir dos archivos de texto, el primero llamado Telefonos.in (figura 3). El que contendrá en la primera línea la cantidad de contactos que posee el archivo, luego en las siguientes líneas se encuentra el nombre completo del contacto (dos nombres y dos apellidos), seguido de su número de teléfono. El segundo archivo tiene por nombre Buscados.in (figura 4). El que contiene en su primera línea cuantos números de teléfono se están buscando y en las siguientes el nombre completo del contacto a buscar.

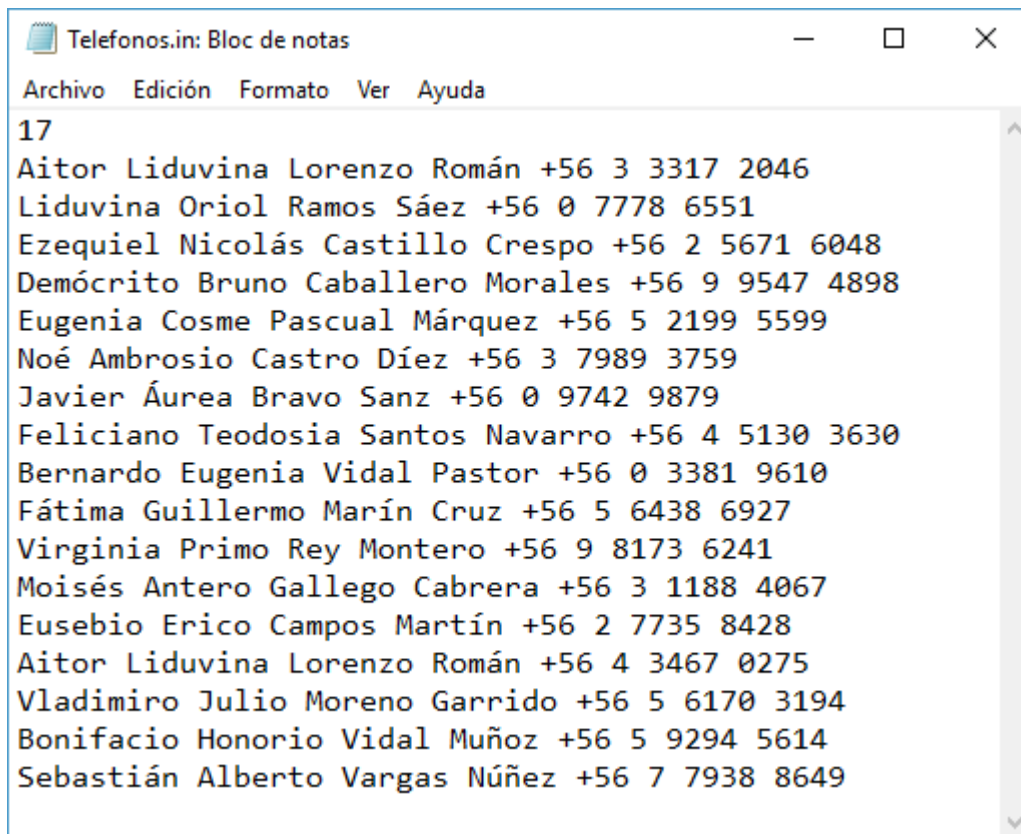


Figura 3: Archivo de entrada Telefonos.in.

Laboratorio 5

Todas las coordinaciones

FACULTAD DE INGENIERIA
Departamento de Ingeniería Informática
Análisis de algoritmos y estructuras de datos



24/05 /2017

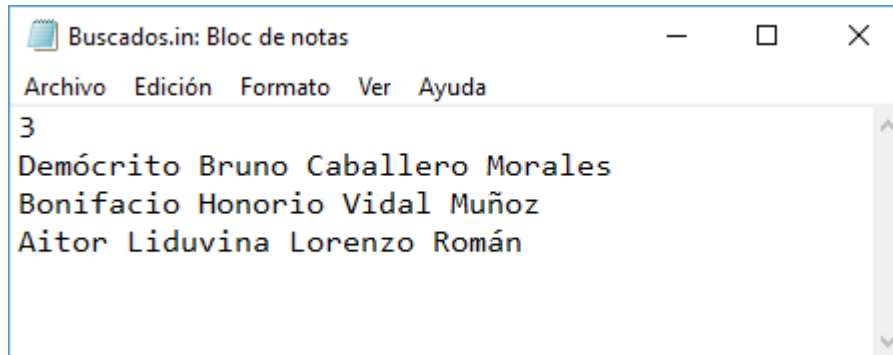


Figura 4: Archivo de entrada Buscados.in.

Como resultado debe generar un tercer archivo de texto llamado TelefonosEncontrados.out (figura 5), que debe contener en nombre completo del contacto y su número de teléfono. En caso de que el contacto aparezca dos veces con números telefónicos distintos, en el archivo TelefonosEncontrados.out debe aparecer el nombre del contacto seguido de los dos números separados por un guión.

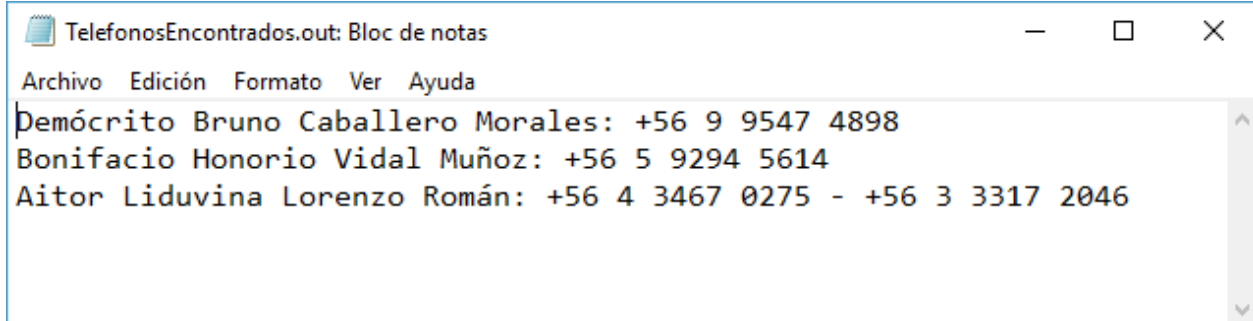


Figura 5 Archivo de salida.

Además, se solicita que el programa pueda mostrar (en otro archivo o por pantalla) una lista con todos los nodos junto con la siguiente información:

- Identificador completo del nodo
- Nivel del nodo
- Identificador completo del hijo izquierdo del nodo
- Identificador completo del hijo derecho del nodo
- Identificador completo del hijo central del nodo (solo en caso de usar árbol 2-3)

Laboratorio 5

Todas las coordinaciones

FACULTAD DE INGENIERIA
Departamento de Ingeniería Informática
Análisis de algoritmos y estructuras de datos



24/05 /2017

Fecha de entrega: 22/06/2017 hasta las 23:30 hrs.

Instrucciones de entrega:

- **Archivo PDF con el informe, incluye:**
 - Comparación entre los algoritmos realizados en los laboratorios 4 y 5.
 - Manual de usuario.
 - Gráfica de rendimiento (cantidad de datos vs tiempo de ejecución).
- **Se debe calcular el $T(n)$ y el O .**
- **Código fuente en archivos .c y .h (no entregar proyectos de ninguna IDE).**
- **El código debe permitir ser compilado en ambiente Windows y Linux por lo que se sugiere usar ANSI C.**