

Universidad de Costa Rica

Computación e Informática

Programación II

Fuga en la tubería

Análisis y Diseño

Gabriel Bogantes Armijo B71146

Profesor: Luis Gustavo Esquivel

Fecha: 26/11/2019

II-Ciclo 2019

El siguiente documento será utilizado para realizar un análisis del problema planteado para la tarea programada uno. Además, se hará un diseño del programa y la solución del problema creando un algoritmo.

Análisis:

Primero, se debe entender el problema a la perfección para poder realizar el diseño adecuado:

Se quiere simular fugas de una tubería. Cada tubería se divide en varias partes y cada parte tiene una forma que permite el ingreso de fluido en cualquiera de las direcciones (norte, sur, este, oeste). Dependiendo de la forma que sea tiene sus direcciones permitidas. El objetivo del proyecto es permitir el ingreso y salida del fluido sin problema alguno (pérdida del fluido). Para solucionar el problema de las fugas, está la ventaja de que para cada posición de las formas puede haber una rotación para ir tapando las irregularidades.

Asimismo, se debe tomar en cuenta que debe haber una entrada y una salida del fluido. Las entradas y las salidas pueden venir en cualquiera de las direcciones válidas (Norte, Sur, Este u Oeste). Pueden haber varias entradas y varias salidas, el usuario lo indica a la hora de ejecutar el programa.

Diseño:

Para crear una solución, se puede recurrir a la modulación de los diversos problemas que hay y buscar soluciones para cada una de las partes. Por ejemplo: Lectura de datos, validar si hay fugas, arreglar las fugas, convertir los datos, manipulación de argumentos, etc. A continuación, se explicará la forma de resolver cada uno de los problemas:

Manipulación de argumentos: Para resolver el problema de manipulación de argumentos hay que saber las diferentes y posibles combinaciones de los mismos: se ha decidido ver todos los casos posibles, escribirlos y realizar una forma de que los argumentos leídos llamen a las subrutinas pertinentes para que sean ejecutadas.

Lectura de datos: Para leer los datos, hay que entender que se puede hacer de dos maneras: por lectura de archivo o digitados en consola. Primero hay que leer y guardar la cantidad de filas, columnas, hidrantes y desagües. Luego hay que leer las posiciones en las que se encuentran los hidrantes y los desagües. Después, se utilizarán la cantidad de filas y de columnas para leer los datos que simulan la tubería (esto ya que se sabe cuántas veces hay que leer los caracteres).

Validar (validate): En este módulo, se pide verificar si la tubería tiene alguna fuga o no, además de decir dónde están las fugas. Para realizar una solución a esto, se toma la decisión de recorrer la tubería e ir revisando pedazo casilla por casilla si a su alrededor hay una tubería con la dirección contraria válida. Si no tiene fugas, no debe hacer cambio alguno, mientras que si tiene fugas hay que imprimir las posiciones de cada una de las fugas y la dirección donde hay fugas.

Arreglar las fugas (solve): Para resolver las tuberías, se seguirá la práctica de la programación llamada “backtracking”. La misma consiste en verificar cada una de las posibles combinaciones. La forma en la que se aplica el backtracking en este caso es ir probando todas las rotaciones que se pueden hacer en todos los espacios y comparando hasta que se encuentre una solución. Asimismo, se debe tomar en cuenta que pueden haber muchas soluciones y que con el backtracking se va a tomar solo la primera que se encuentra.

Convertir (convert): Para este problema, se pide que el programa sea capaz de interpretar datos en binario y en formato .txt. Para solucionarlo deben haber dos subrutinas, una para convertir de texto a binario y una para convertir de binario a texto.