



Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencias de la Computación

Informe Tarea 0

IIC2133 – Estructura de datos y algoritmos

Segundo semestre, 2017

Nombre: Daniel Soto Silva

Número de alumno: 14206153

Descripción del programa

En este programa, el cual consiste específicamente en simular la evolución de un ecosistema habitado por bacterias a través de distintas generaciones (paso del tiempo).

Para esta simulación, el programa en general se puede resumir en los siguientes algoritmos principales:

- **Lectura de archivos y poblamiento del ecosistema:** En este algoritmo, se basa principalmente en la lectura de la información entregada en el archivo, el cual es dado como input en la línea de comandos, junto con la cantidad de generaciones que se simulará. Durante esta acción, en primer lugar se lee la primera línea del archivo el cual contiene el número de filas y columnas de la grilla que emula al ecosistema, dichos valores son guardados posteriormente en sus respectivas variables. Inmediatamente mediante un ciclo "for" se procede a leer las siguientes líneas del archivo que contienen la información de las casillas con y sin bacterias, para luego actualizar en cada iteración de lectura la casilla de la matriz (arreglo multidimensional), mediante otro ciclo for anidado al primero, creada previamente como la principal estructura de datos, que contiene dicha información. Para hacer la lectura más sencilla, se utilizó el comando "fscanf".
- **Verificación de condiciones de sobrevivencia para cada casilla:** En este algoritmo, se basa principalmente en recorrer la matriz que contiene la información de cada casilla de la grilla en la que está la información de si hay o no bacterias. Para este recorrido se ocupan básicamente "2 ciclos for anidados" (ya que debemos si o si chequear, mediante controles de flujo como el if, y el else if, las condiciones de cada casilla respecto a la población de su alrededor, ya que estas condiciones definirán si habrá o no alguna bacteria viva en la siguiente generación). Durante esta misma verificación se actualiza la información en la interfaz gráfica para actualizar la visión de esta (mediante la función `watcher_refresh()`). Se resalta que esta verificación se hace las veces que indique la cantidad de generaciones dada como parámetro al inicio del programa, las cuales serán controladas mediante un ciclo for que anidará a los otros dos ciclos anteriores que recorren la matriz.



Análisis de algoritmo

Para hacer el análisis de algoritmo, se centrará específicamente en los algoritmos descritos en el apartado anterior ya que básicamente se llevan el total de las acciones realizadas en el programa.

Para el algoritmo que realiza la lectura del archivo dado como parámetro y posteriormente poblar la matriz (arreglo multidimensional) creado para almacenar la información de las casillas de la grilla, prácticamente los pasos que realiza (o las iteraciones que se realiza) dependerá de la cantidad de filas y columnas de esta grilla, por lo tanto al ser dos ciclos for anidados para realizar esta acción la complejidad del algoritmo en el peor de los casos (que esta matriz que representa la grilla sea cuadrada) es $O(n^2)$.

Para el algoritmo que realiza la verificación de las condiciones de sobrevivencia y posterior actualización de esta información en la interfaz gráfica, básicamente se realiza dos ciclos for anidados en las cuales se preocupa de recorrer cada casilla de la matriz que representa a la grilla del ecosistema de bacterias, los pasos que se realizan dependerán de la cantidad de filas y columnas dado como parámetros (la cual se obtuvo previamente en el mismo archivo al hacer una lectura a la primera línea), en el peor de los casos (que esta matriz sea cuadrada) la complejidad del algoritmo es $O(n^2)$. Por otro lado en este análisis no estamos considerando los pasos o iteraciones realizadas que representan la cantidad de generación que se está simulando, lo cual en un principio es dado al programa como parámetro. Podríamos decir que considerando estas iteraciones que anida a los dos ciclos for antes descritos, la complejidad del algoritmo sería $O(n^3)$, pero esto sería limitar a que el parámetro guarde relación con el número de filas y columnas de la grilla, cosa que no es así ya que son datos independientes, por lo que es más correcto decir que la complejidad del algoritmo, dado un número de iteraciones "m" que representa la cantidad de generaciones en la simulación, es $O(m * n^2)$.

Analizando el uso de memoria utilizado por el programa, sin considerar el que ocupa las variables básicas del tipo int, char, que fueron los ocupados en el programa, lo que se llevó la mayor parte de la memoria ocupado por esta fue el arreglo multidimensional creado, en la cual cada casilla es del tipo int, por lo cual la memoria ocupada por cada casilla de la matriz va a depender de la cantidad de bits asignado para las variables del tipo int (lo cual puede variar, dependiendo de la arquitectura propia del computador, por ejemplo asignar 4 bytes a este tipo de variable) y junto a esto, la memoria total ocupada por la matriz también dependerá de la cantidad de filas y columnas indicadas en los archivos dado como parámetros.