ESTRUCTURA DE DATOS Y ALGORITMOS 2

INFORME AVANCE SEMANA

PROYECTO #1 – 11208 AIRPLANE SCHEDULING

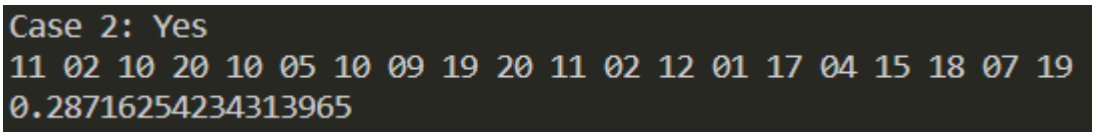
PROFESOR: JUAN GUILLERMO LALINDE PULIDO

KRISTIAN RESTREPO OSORIO

EAFIT

2023

Para la anterior semana, presentábamos el problema de que el algoritmo cumplía todos los casos de “UDEBUG”, pero dos casos en específicos se demoraban mucho para retornar la solución, lo cual no cumple con el tiempo limite establecido. Ahora, el algoritmo devuelve el resultado de todos los aeropuertos de los casos de prueba de “UDEBUG” en un tiempo satisfactorio, el caso que mas se demora, es el caso que anteriormente demoraba 15 min, ahora, demora tan solo 0.2 segundos, como se muestra en la imagen presentada a continuación:



Se realizaron pruebas de escritorio validando que el algoritmo establece los parqueaderos adecuados a cada uno de los aviones.

Los cambios que se realizaron al código con respecto a la anterior entrega fueron:

- En la función “ponderar”, a la lista de visitados definida anteriormente, la modificamos como un conjunto que guarda las coordenadas de las casillas que ya fueron ponderadas. Esto porque, después de investigar un poco, entendimos de los conjuntos que están diseñados para proporcionar una búsqueda rápida de elementos, ya que Python utiliza una tabla de hash para encontrar el elemento (que tiene complejidad de O(1)), lo cual era una mejor implementación que tener una lista para los visitados, ya que para una lista se debe hacer una búsqueda lineal del elemento, con lo que puede llevar mas tiempo a medida que la lista crecía.
- En base a la búsqueda anterior, se me ocurrió también que podía almacenar los parqueaderos en un conjunto, y a la hora de asignar un parqueadero, no recorrer toda la matriz buscando alguno disponible, sino recorrer los elementos de dicho conjunto. Así, se vuelve una búsqueda lineal, eliminando el doble ciclo que teníamos para recorrer la matriz y encontrar un parqueadero.
- En la función “resolver_problema”, la que se encarga del backtracking, antes de revisar si es un evento de aterrizaje o despegue, se cuenta cuantas entradas consecutivas de aviones hay, si hay mas entradas de aviones consecutivas que parqueaderos disponibles, retorna falso, ya que esa no será una solución factible y se debe buscar otra posibilidad. Esto permite minimizar en gran medida los llamados recursivos que se hacen a la hora de asignar parqueaderos.

En el jurado de “ONLINE JUDGE”, el algoritmo no cumple el tiempo limite para los casos de prueba a los que se les busca una solución. Aun se está trabajando en darle una solución a lo mismo, ya que en los casos de prueba que tenemos, los cumple todos en un tiempo realmente bueno, pero no se entiende aún el porque el jurado retorna que excede el tiempo límite.