

ESTRUCTURA DE DATOS Y ALGORITMOS 2

ALGORITMOS GREEDY

PROYECTO #2 – 11208 AIRPLANE SCHEDULING

PROFESOR: JUAN GUILLERMO LALINDE PULIDO

KRISTIAN RESTREPO OSORIO

EAFIT

2023

¿Cuál es la decisión que se toma que nunca se revisa?

La decisión que se toma y nunca se revisa es **el lugar en donde se va a ubicar el avión**, ya que se desea establecer el mismo en el parqueadero que menos caminos obstruya, para esto, utilizamos una función la cual calcula el numero de caminos que se ven afectados si un avión se establece en cualquiera de los parqueaderos. También recalcamos, que recorremos los parqueaderos de menor ponderación a mayor ponderación comprobando si dicho parqueadero es el que afecta menos caminos, lo hacemos de esta forma, ya que es mas optimo parquear el avión en un lugar que tenga el mínimo de ponderación (ya que permite que el avión tenga más posibilidades de salir) y sea el que menos caminos afecte parquear un avión en esa posición. Por lo tanto, el avión se establece en el parqueadero que afecte el mínimo de caminos. (como lo mencionamos anteriormente, si el mínimo de caminos lo tiene un parqueadero con una ponderación baja, lo establece allí, ya que recorre los parqueaderos en orden ascendente de ponderación).

¿Cuál es el criterio utilizado para seleccionar la siguiente decisión?

Como se explicaba anteriormente, el criterio para seguir parqueando es que el lugar donde vamos a ubicar el avión sea el que menos caminos afecte, y como recorremos los parqueaderos de menor a mayor ponderación, si un parqueadero diferente al de mayor ponderación afecta el mínimo de caminos, esta será la posición más adecuada para ubicar el avión, puesto que se tiene que pasar por menos parqueaderos para llegar al mismo, y con esto, es mas probable que aviones en un futuro no bloqueen su camino, como pasaría si solo lo ubicáramos en el de mayor ponderación.

Si no es posible aterrizar o despegar un avión, la lista de eventos inicial se modifica, y el evento actual se lleva al final de la lista de eventos, para poder darle al usuario una respuesta aproximada a la solución definitiva del problema.

Compare la respuesta obtenida por el algoritmo codicioso con la respuesta exacta

El algoritmo, para muchos casos, retorna la respuesta valida, tomando como ejemplo los casos de prueba de UDEBUG, todas las matrices pequeñas que tenían una solución, y las matrices 20,17,12,11,10,9,8,3 de los casos de prueba grandes, dan una solución correcta y exacta en base a la lista de eventos ingresada. Ahora bien, existe una diferencia cuando la decisión greedy que tomamos, no nos permite llegar a una solución exacta, ya sea porque un avión no puede parquear, o no puede despegar una vez ya aterrizado, en este caso, para darle una respuesta aproximada al usuario sobre una solución que se le podría dar al problema, ya que al ser greedy no vamos a devolvernos y reubicar los aviones, será que dicho evento lo aplazamos para el final de la lista de eventos actual, así, permitimos que el resto de eventos se sigan llevando a cabo con normalidad, para finalmente, ubicar y/o despegar dicho avión problemático y se pueda llegar a una solución aproximada. Por lo tanto, nuestro algoritmo puede imprimir tres respuestas:

- La respuesta valida, donde en base a la lista de eventos otorgada, y la propiedad con la que establecemos los aviones en el parqueadero mas optimo para que el mismo pueda aterrizar y despegar satisfactoriamente, se pudo llegar a una respuesta correcta para esa lista de eventos.

```

14 8 6
## .. ## .. 16 17
05 .. ## ## 13 11
## 01 ## 03 ## ##
## ## 02 ## .. 06
## .. 08 12 .. ##
## 14 .. 04 == ##
.. .. 07 ## ## 09
15 .. .. ## == 10
+14 +13 -13 +8 +7 +2 +4 +6 -4 -2 +9 -6 -14 -7 +3 +12 +10 -3 +1 -9 -12 -8 +5 -5 +11 -1 -10 -11
Case 20: Yes
12 15 12 06 07 10 09 07 15 06 07 09 07 12

```

- La respuesta aproximada, donde no se pudo llegar a una solución en base a la decisión que se toma de donde establecer los aviones, por lo tanto, se aplaza dicho evento problemático para el final de la lista de eventos para que se pueda seguir llevando a cabo el resto de los eventos, y que finalmente, cuando no haya aviones “estorbando”, dicho evento pueda llevarse a cabo y se dé una respuesta aproximada al usuario. Por lo tanto, se informa que, si la lista eventos cambia de esa nueva manera, se podría llegar a dicha solución aproximada.

```

20 10 10
== .. .. .. ## 09 10 11 ##
.. .. .. ## 08 ## 12 ##
.. .. .. ## 07 ## 13 ##
.. .. .. ## 06 ## 14 ##
.. .. .. ## 05 ## 15 ##
.. .. .. ## 04 ## 16 ##
.. ## ## ## 03 ## 17 ##
.. .. .. ## 02 ## 18 ##
.. .. .. ## 01 ## 19 ##
.. .. .. ## 20 ##
+1 +2 +3 +4 +5 +6 +7 +8 +9 +10 +11 +12 +13 +14 +15 +16 +17 +18 +19 +20 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 -11 -12 -13 -14 -15 -16 -17 -18 -19 -20
Case 1: No
Si se modifica la lista de eventos de la siguiente forma:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 -20 -19 -18 -17 -16 -15 -14 -13 -12 -11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1
Obtenemos la siguiente solución:
20 19 18 17 16 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01

```

- Cuando no existe posibilidad, es decir, cuando el aeropuerto ingresado no tiene parqueaderos, pues es imposible llegar a una solución en la misma, y no se puede modificar la lista de eventos para llegar a una solución aproximada porque ningún evento puede llevarse a cabo directamente.

```

4 4 3
== .. ..
.. ## ..
.. ## ..
.. ## ..
+1 +2 +3 -3 +4 -2 -1 -4
Case 1: No

```

Al estar abordando un problema de fuerza bruta y backtracking con un algoritmo codicioso, se busca llegar a la respuesta mas aproximada posible de la solución problema. Por lo tanto, para los casos en que reubicamos el evento para dar una respuesta aproximada, pues esta será notoriamente distinta a la exacta, por el mismo cambio que se le realiza a la lista de eventos original.