

# INFORME. IMPLEMENTACIÓN DEL ALGORITMO BACKTRACKING EN LA PROBLEMÁTICA ‘AIRPLANE SCHEDULING’.

Nombre de la estudiante: Manuela Caro Villada

Código: 1000124254

Correo: [mcarov@eafit.edu.co](mailto:mcarov@eafit.edu.co)

Materia: Estructura de Datos y Algoritmos II

## 1. PROBLEMÁTICA:

El problema de la asignación de estacionamiento surge en el contexto de los aeropuertos gestionados, donde a cada avión que aterriza se le debe asignar un espacio de estacionamiento y garantizar que se cumplan ciertas restricciones. Este problema se puede plantear como un desafío de optimización combinatoria, cuyo objetivo es encontrar una solución factible que maximice el uso de los espacios de estacionamiento y permita que todas las aeronaves aterricen y despeguen de manera segura.

## 2. SOLUCIÓN:

La siguiente explicación que se brindará igualmente es errónea porque el código no funciona, pero se desea dar a entender un contexto de la idea aunque no haya encontrado una explicación a su fallo.

### 2.2 ALGORITMO:

El algoritmo utiliza un método alternativo para asignar espacios de estacionamiento a las aeronaves. Primero comprueba que la jugada es válida mediante la función ‘is\_valid\_move()’, que comprueba que las coordenadas están dentro del damero, que la celda correspondiente no está ocupada por un plano ('##'), y que no ha sido asignado previamente. A continuación, se llama recursivamente a la función ‘backtrack()’ para cada evento. Si el evento es positivo, se busca un espacio vacío ('..') en el tablero y se asigna un plano a ese espacio. Si el evento es negativo, se busca el intervalo de tiempo asignado ('==') y se libera la aeronave. Realice una llamada recursiva en el próximo evento e invierta la asignación o des-asignación después de la llamada.

### 2.3 REPRESENTACIÓN:

Para dejar claro cómo se representa la información en el algoritmo de backtracking, se proporciona una tabla que describe todos los datos de entrada y salida:

Descripción del dato	Tipo de dato	Representación en el algoritmo	Conversión
Número de filas	Entero	rows	Directa
Número de columnas	Entero	cols	Directa
Tablero	Matriz de cadenas	flat_land	Directa
Eventos	Lista de enteros	events	Directa

## 2.4 ASIGNACIÓN DE PARQUEADEROS:

El proceso de asignación de un espacio de estacionamiento es el siguiente:

- a) Los criterios para seleccionar la siguiente aeronave a asignar son comprobar la secuencia de eventos y comprobar si el evento es positivo (aterrizaje) y aún no se ha asignado un parqueadero para ese avión.
- b) El criterio para elegir un lugar de estacionamiento es atravesar las celdas del tablero de ajedrez de acuerdo con la función 'is\_valid\_move()' y encontrar un espacio ('..') que coincida con la condición de movimiento válido.
- c) La comprobación de si la plaza de aparcamiento está libre se realiza mediante la función 'is\_valid\_move()', que comprueba si las coordenadas de la plaza de aparcamiento son válidas y si la celda correspondiente está libre.
- d) En caso de no encontrar un parqueadero disponible para asignar al avión, se realiza la retrotracción (backtracking) y se desasigna el parqueadero previamente asignado

## 2.5 VALIDACIÓN DE DESPEGUE:

La prueba de arranque se realiza realizando los siguientes pasos:

- a) Determinar si la aeronave puede despegar creando una ruta desde el puesto de estacionamiento designado hasta el lugar de despegue. Se recorre el tablero desde la posición del plano hasta el punto de partida, comprobando cada celda para ver si está libre ('..') y si es un movimiento válido según la función 'is\_valid\_move()'.
- b) Si la aeronave no puede despegar por celdas ocupadas o movimiento inválido, realizar un regreso y recordar las tareas anteriores.

## 3. RECOMENDACIONES:

- a) Se sugiere transformar los datos de entrada en una representación adecuada para simplificar el proceso de resolución de problemas.
- b) Aunque este problema se plantea como asignación de estacionamiento matricial, se puede abordar como un problema gráfico donde los nodos de llegada y salida se marcan con '==', los nodos vacíos '..' y los obstáculos '##' se pueden descartar.
- c) Es importante tener en cuenta que el lugar de aterrizaje de la aeronave puede ser diferente al lugar de salida, por lo que se deben asignar suficientes espacios de estacionamiento para cada evento.
- d) Las coordenadas solo son necesarias para la construcción del grafo y no se necesitan después de eso.

## 4. COMPONENTES TÉCNICOS:

- a) El lenguaje de programación es Python debido a su facilidad de uso y amplio soporte para estructuras de datos y algoritmos.
- b) No se define ninguna clase en el código dado.
- c) La función definida es:

‘is\_valid\_move(x, y)’: Comprueba si la coordenada es un movimiento válido.  
‘backtrack(index)’: Realiza la asignación de estacionamiento utilizando el método de retroceso.

Ambas funciones definen parámetros de entrada, tipos de datos de retorno y excepciones.

- d) No se definen otras variables en el código adjunto. Este informe proporciona una descripción clara del algoritmo de asignación de estacionamiento utilizando el proceso de rastreo, representación de datos, asignación inicial y verificación. También incluye recomendaciones e información técnica sobre la implementación.

## 5. CONCLUSIONES:

El problema de asignación de espacios de estacionamiento resuelto mediante algoritmos de backtracking es un desafío complejo que requiere un análisis cuidadoso y la consideración de las restricciones del problema. Al desarrollar un informe, es posible comprender en detalle cómo funciona el algoritmo y los pasos necesarios para encontrar una solución real. Durante el análisis del código entregado, se identificaron algunas razones por las que no funcionaba completamente:

Error de validación de movimiento: el código implementa la función is\_valid\_move(x, y) para verificar si la posición (x, y) es un movimiento válido. Sin embargo, el código contiene errores en la lógica de validación que pueden causar que los espacios de estacionamiento se asignen incorrectamente.

Ignorar casos especiales: el código ignora los casos especiales en los que no se pueden encontrar asignaciones de estacionamiento válidas para todas las aeronaves. En este caso, el algoritmo debe proporcionar un mensaje apropiado o manejar la situación correctamente.

En general, se puede concluir que el código presentado debe cambiarse y mejorarse para implementar correctamente su función: la asignación de espacios de estacionamiento de manera óptima y dentro de los límites del problema.