



# Algoritmos de Búsqueda: Ejercicio 5 y 6

Barraquero Ignacio, Campo Camila, Villarreal Francisco, Marzari Agustina  
Facultad de Ingeniería



## Introducción a los Algoritmos de búsqueda

los algoritmos de búsqueda son métodos sistemáticos para explorar un espacio de estados con el objetivo de encontrar una solución a un problema. Dicho espacio puede representarse como un árbol o grafo de búsqueda, donde los nodos representan estados posibles y los arcos representan acciones que permiten pasar de un estado a otro. En este trabajo se analizarán y compararán cuatro enfoques representativos: búsqueda en profundidad, búsqueda en profundidad con límite, búsqueda avara y búsqueda A\*, aplicados al problema de encontrar un camino en un tablero desde un estado inicial hasta un estado objetivo.

## Búsqueda por profundidad

- **Estrategia:** expande siempre el último nodo generado (usa pila LIFO).
- **Recorrido:** se mete lo más profundo posible en una rama antes de retroceder.
- **Ventajas:**
  - Muy bajo consumo de memoria (solo guarda el camino actual).
  - Sencillo de implementar.
- **Desventajas:**
  - No garantiza encontrar el camino más corto ni el de menor costo.
  - Puede quedarse “atrapada” en ramas infinitas si el grafo es grande o infinito.

## Búsqueda en profundidad con Límite

- **Estrategia:** es DFS pero con una profundidad máxima establecida.
- **Motivación:** evita que DFS se quede atrapada explorando infinitamente.
- **Ventajas:**
  - Evita ciclos infinitos.
  - Controla hasta dónde explorar.
- **Desventajas:**
  - Si el límite es muy bajo, puede no encontrar la solución.
  - Si el límite es muy alto, pierde eficiencia.

## Búsqueda Avara

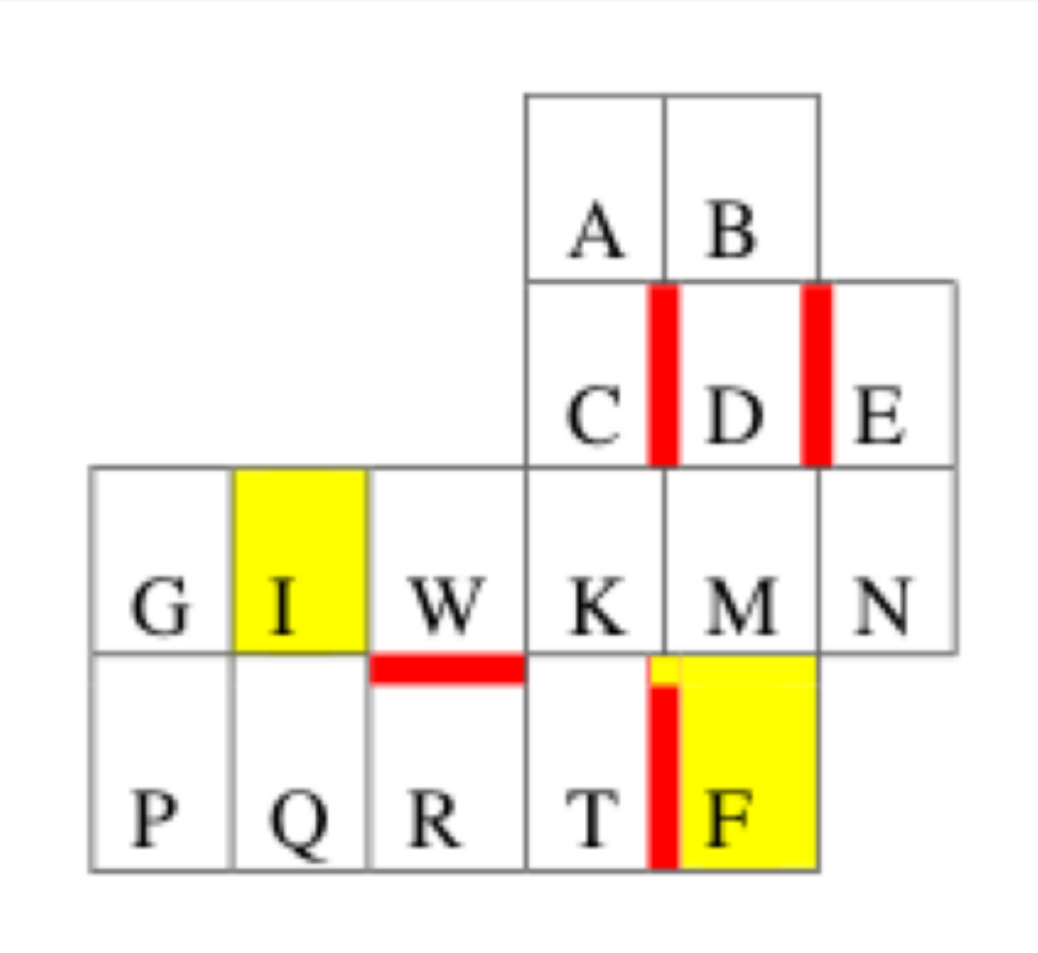
- **Estrategia:** expande siempre el nodo que parece más cerca de la meta según una heurística  $h(n)$
- **Ventajas:**
  - Suele ser rápida, porque se guía “directo” hacia la meta.
  - Usa menos memoria que A\*.
- **Desventajas:**
  - No garantiza encontrar el camino más corto.
  - Puede elegir un camino muy caro (ejemplo: pasar por W con costo 30).

## Búsqueda A\*

- **Estrategia:** combina costo real y heurística.
- **Ventajas:**
  - Garantiza encontrar el camino más barato si la heurística es admisible (no sobreestima el costo real).
  - Es más informada y eficiente que una búsqueda exhaustiva.
- **Desventajas:**
  - Puede requerir mucha memoria (guarda todos los nodos abiertos).
  - Más costosa computacionalmente que Greedy o DFS.

## Ejercicio 5

Se busca que el algoritmo vaya desde la casilla I a la F, aplicando los tres métodos anteriores. No se puede pasar por casillas que tengan una pared(rojo), y el movimiento sigue la regla de Manhattan. Para el A\*, la casilla W cuesta 30, las demás cuestan 1.



La resolución muestra como con distintos algoritmos se obtienen distintas trayectorias

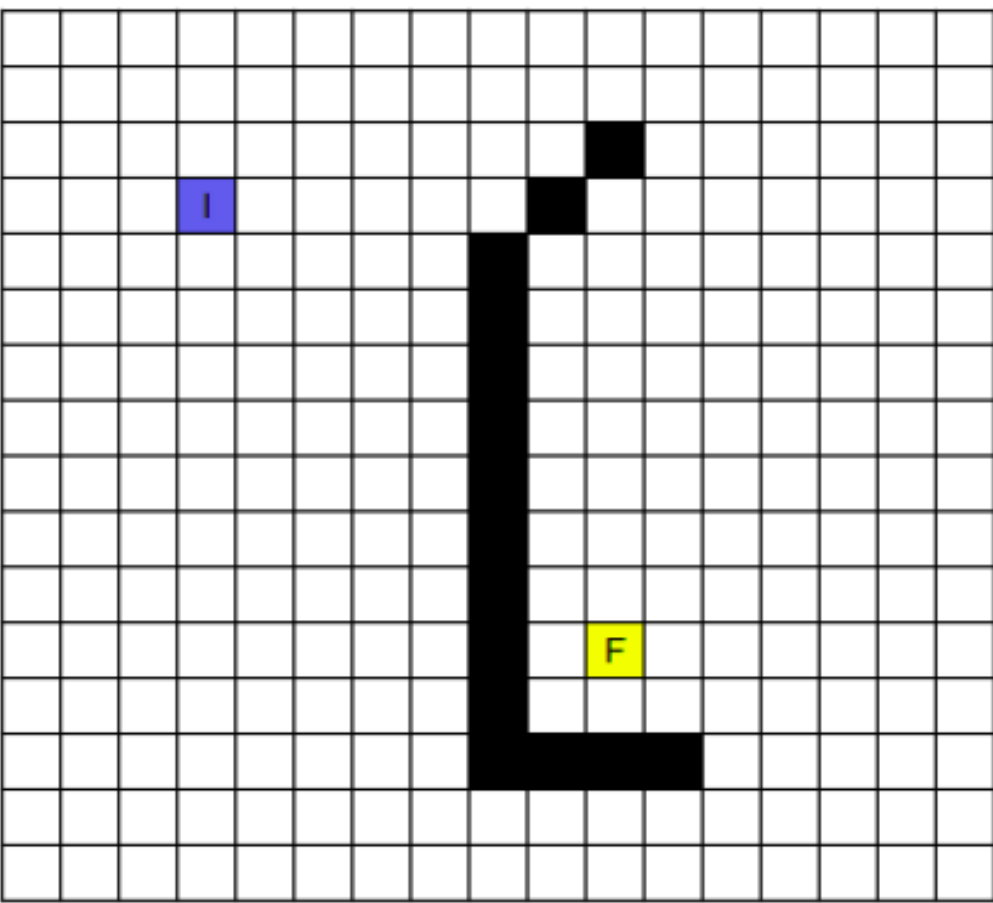
```
DFS : ['I', 'Q', 'P', 'Q', 'R', 'T', 'K', 'M', 'F']
Elige el camino por profundidad y en caso de empate por alfabeto

Greedy: ['I', 'Q', 'R', 'T', 'K', 'M', 'F']
Busqueda heurística, prioriza el camino mas corto, en caso de empate elige por alfabeto

A* : ['I', 'Q', 'R', 'T', 'K', 'M', 'F']
Cada casilla tiene un costo Real, prioriza el menor costo, en caso de empate elige por alfabeto costo: 6
PS C:\Users\Camila Campos\Desktop\UNCOMO\IA 1>
```

## Ejercicio 6

Desarrollar un agente que emplee una estrategia de búsqueda A\* para ir de una casilla a otra evitando la pared representada, pudiendo seleccionar ustedes mismos el inicio y el final. Muestre en una imagen el camino obtenido.



Este código utiliza el método A\* explicado anteriormente para encontrar el camino (recuadros celestes) entre un nodo inicial (recuadro azul) y un nodo objetivo (recuadro verde) en medio de los cuales se encuentra una pared (recuadros negros).

