



Pontificia Universidad
JAVERIANA
Bogotá

Inteligencia Artificial

Búsquedas en Espacio del Problema

Ing. Andrea Rueda, PhD
Ing. Enrique González, PhD

Departamento de Ingeniería de Sistemas

Agenda – Búsquedas



1 – Búsqueda a Ciegas

- Profundidad vs Amplitud
- Profundidad Iterativa - Bidireccional
- Taller 2-2: Ejemplo búsqueda a ciegas

2 – Búsqueda con Información

- Función Heurística y Función de Costo
- Métodos de Escalada
- Best First – Costo Uniforme – A*
- Taller 2-3: Ejemplo búsqueda con información

Búsqueda en Espacio Problema

Problemática

- espacios de estados son demasiado grandes
 - imposible representación explícita + no es necesario considerarlos completamente
- generar un árbol de búsqueda
 - representar sólo la parte del espacio de estados que debe ser considerada para resolver el problema

Árbol de Búsqueda

- **Nodos** → Estados
 - Raíz del árbol → estado inicial
- **Arcos** → Operadores
- generar estado sucesores → expandir un nodo conocido
- estrategia de búsqueda → qué estado expandir
 - mecanismo para evitar ciclos → grafo se convierte en árbol
 - nodos conocidos que esperan ser expandidos forman la frontera

Búsqueda en Espacio Problema



Pontificia Universidad
JAVERIANA
Bogotá

					2	8	3						
					1	6	4						
					7		5						

2- Búsqueda a Ciegas

Características

- No requiere de conocimiento específico del problema
 - basta con las definiciones dadas por la definición del problema en el espacio de estados y operadores
- Búsqueda No Informada o de Fuerza Bruta
 - en el peor caso termina recorriendo todo el grafo

Estrategias de Búsqueda a Ciegas

- Profundidad
- Amplitud
- Profundidad Iterativa
- Bidireccional

Búsqueda en Profundidad

Recorrido por Ramas

- exploración exhaustiva de una rama
 - aparecen callejones sin salida

Backtracking

- si se llega a una hoja y no hay solución, entonces regresar hasta el último **nodo con alternativas**

Características

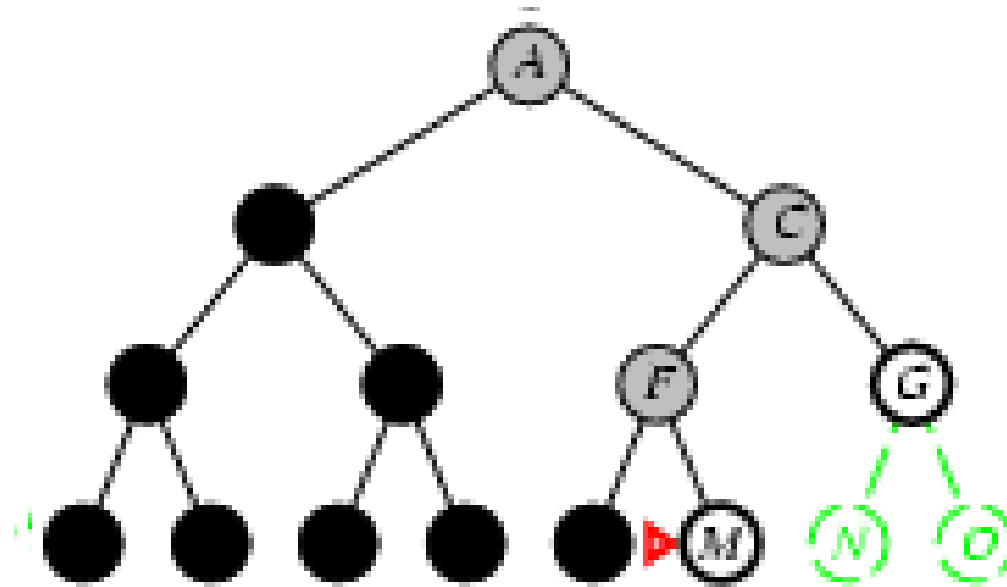
- No siempre es Óptimo
 - puede tomar primero la rama del camino largo
- Completo
 - si no hay ramas infinitas
- No requiere Mucha Memoria
 - manejo de una lista única con el camino recorrido para **evitar ciclos y almacenar la solución**

Búsqueda en Profundidad



Pontificia Universidad
JAVERIANA
Bogotá

Funcionamiento



Búsqueda en Profundidad - Ejemplo

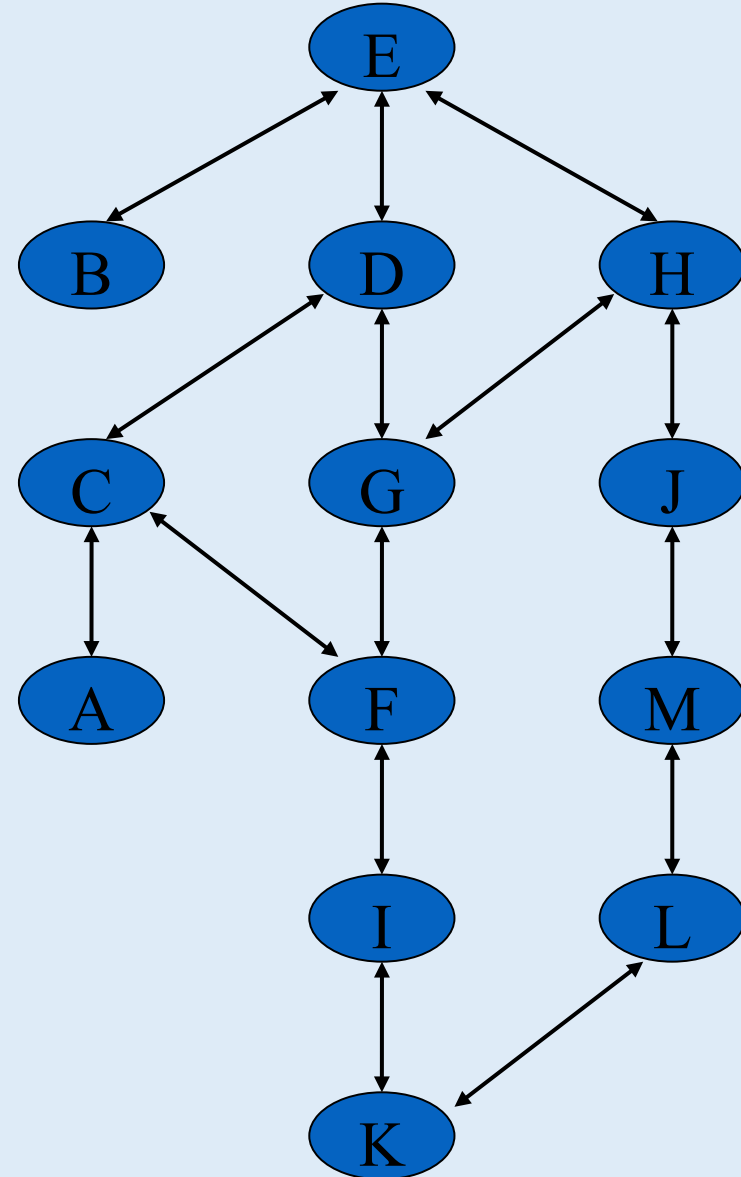


Pontificia Universidad
JAVERIANA

Ejemplo

Planificación de Trayectorias de un Robot

A		B
C	D	E
F	G	H
I		J
K	L	M



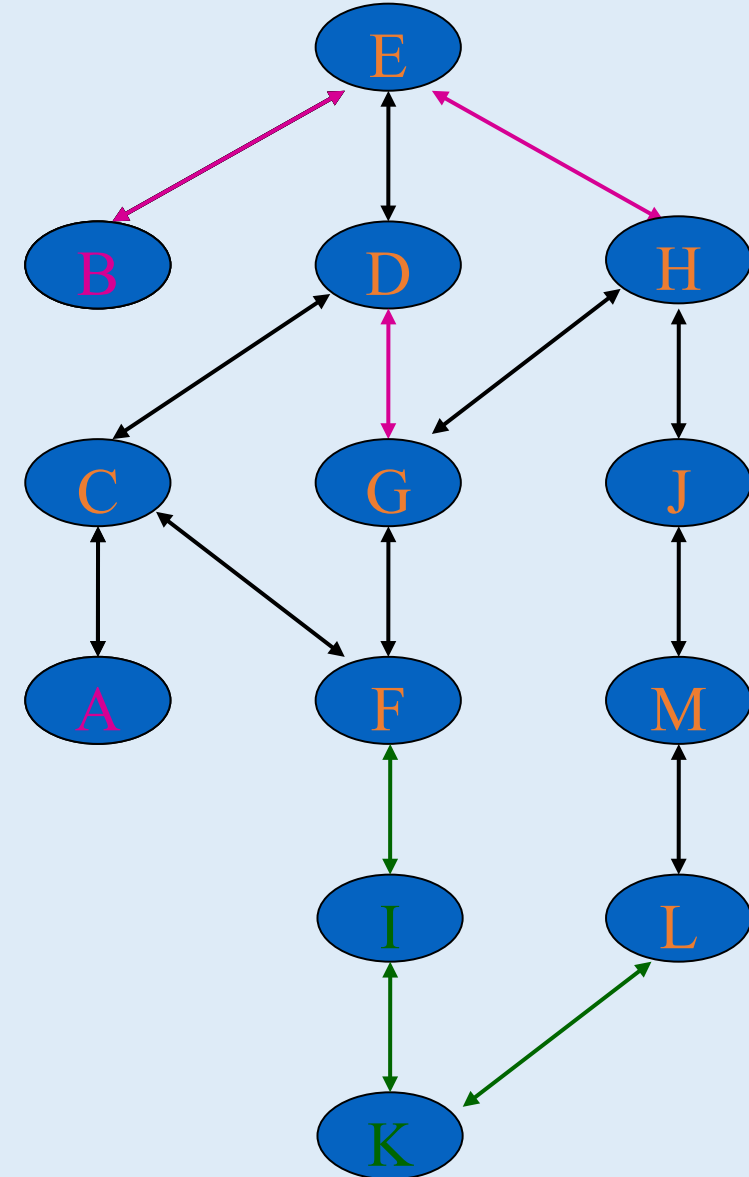
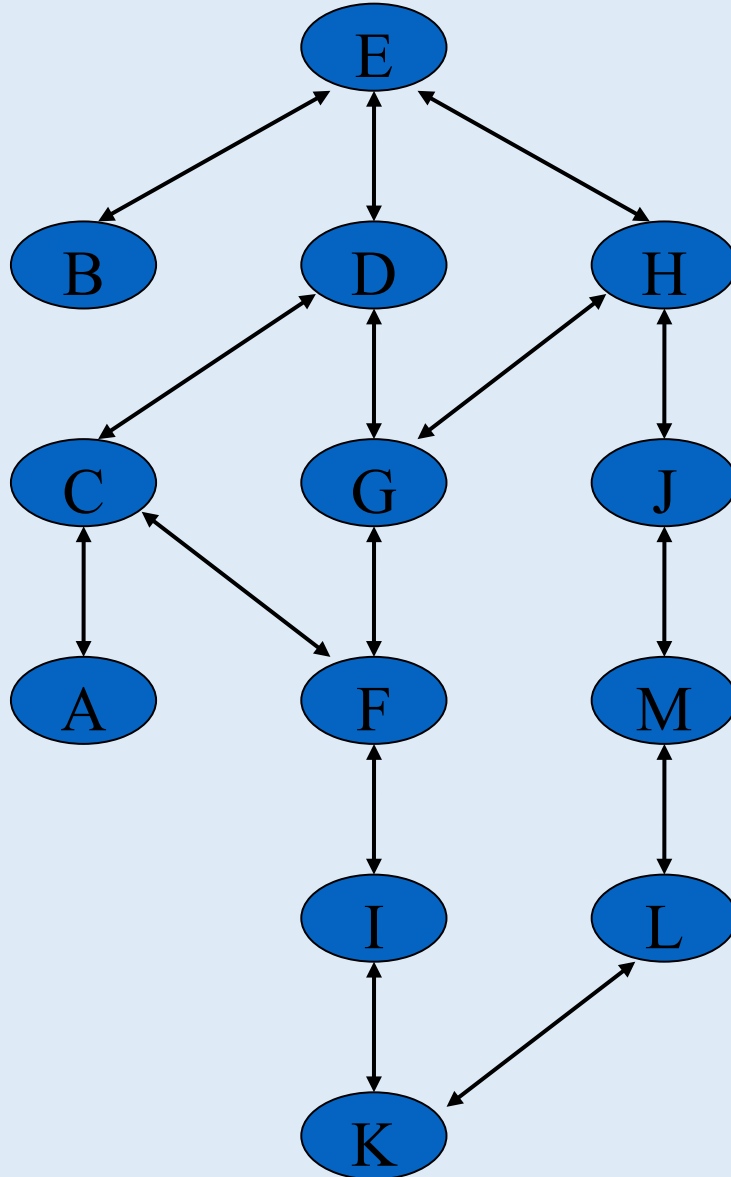
Búsqueda en Profundidad - Ejemplo



Pontificia Universidad
JAVERIANA

Orden
Alfabético

A		B
C	D	E
F	G	H
I		J
K	L	M



Búsqueda en Profundidad - Ejemplo

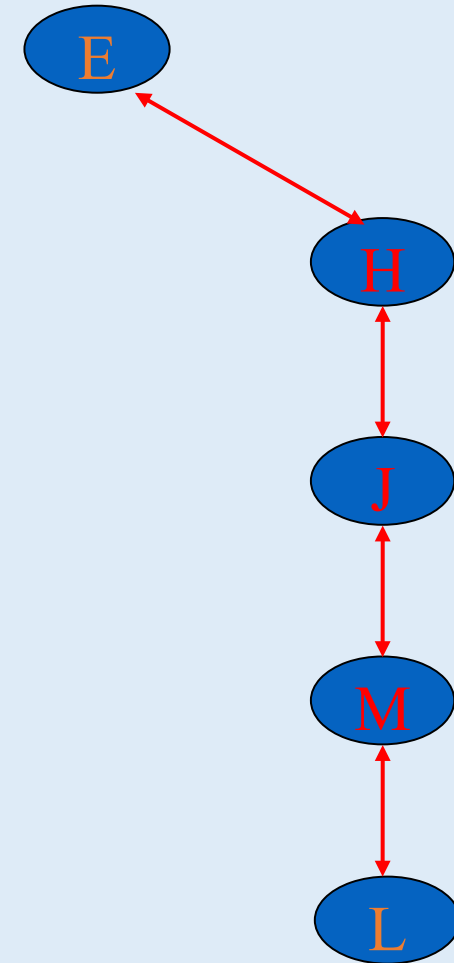
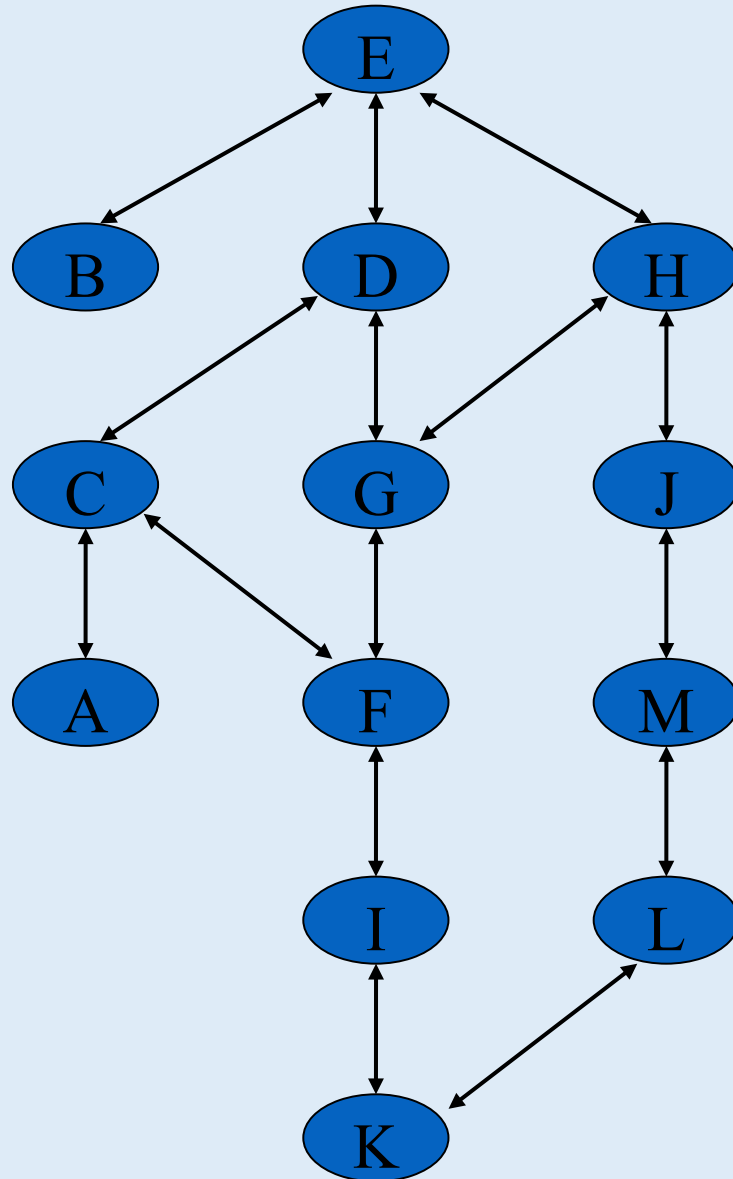


Pontificia Universidad
JAVERIANA

Orden

Contra
Alfabético

A		B
C	D	E
F	G	H
I		J
K	L	M



Búsqueda en Amplitud



Pontificia Universidad
JAVERIANA
Bogotá

Recorrido por Niveles

- se recorren primero todos los hijos, luego todos los nietos, y así sucesivamente
 - no hay callejones sin salida – no requiere *backtracking*

Características

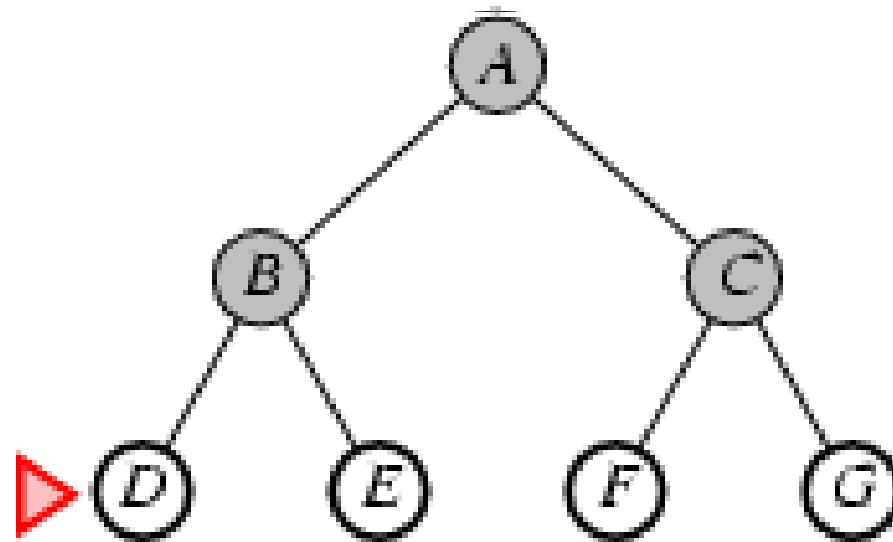
- Siempre es Óptimo
 - encuentra la solución más cercana a la raíz
- Completo
 - garantiza encontrar una solución cuando ésta existe
- Requiere Mucha Memoria
 - manejo de múltiples listas para controlar ciclos y saber el camino hasta ellos
 - nodos conocidos aún no explorados quedan abiertos
 - se requiere una lista por cada nodo abierto

Búsqueda en Amplitud



Pontificia Universidad
JAVERIANA
Bogotá

Funcionamiento



Búsqueda en Amplitud - Ejemplo

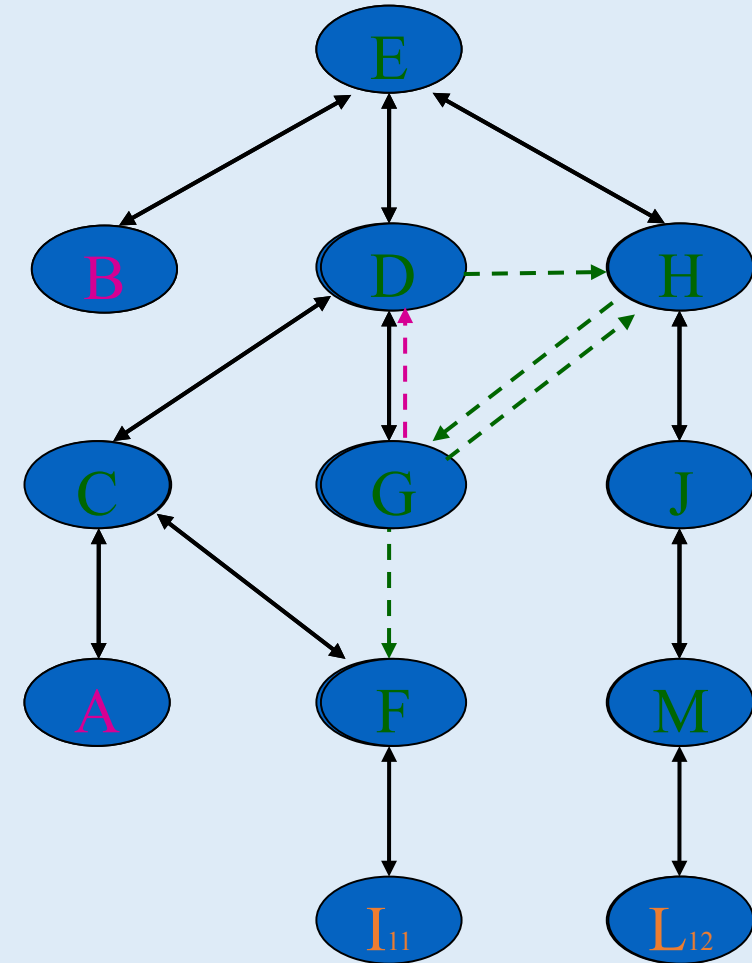
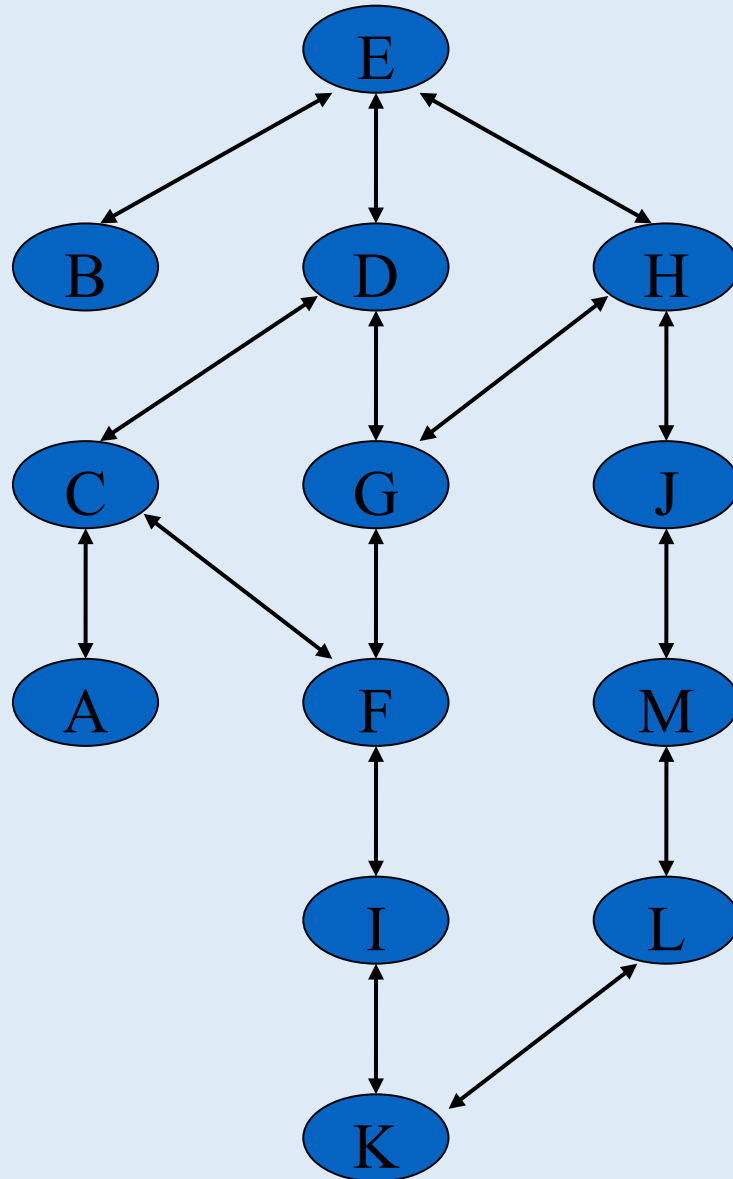


Pontificia Universidad
JAVERIANA

Orden

Contra
Alfabético

A		B
C	D	E
F	G	H
I		J
K	L	M



Búsqueda en Profundidad Iterativa



Pontificia Universidad
JAVERIANA
Bogotá

Basado en Profundidad Limitada

- se exploran las ramas hasta un nivel máximo establecido

Método Iterativo

- si para un nivel de profundidad dado no hay solución, entonces se aumenta el nivel en 1 y se lanza nuevamente la búsqueda profundidad limitada

Características

- Siempre es Óptimo
 - exploración de nuevos nodos sigue un patrón similar al de amplitud
- Completo
 - garantiza encontrar una solución cuando ésta existe
- Requiere Poca Memoria
 - opera igual al de profundidad
 - **overhead** razonable – repite muchas veces los primeros niveles del árbol donde hay pocos nodos

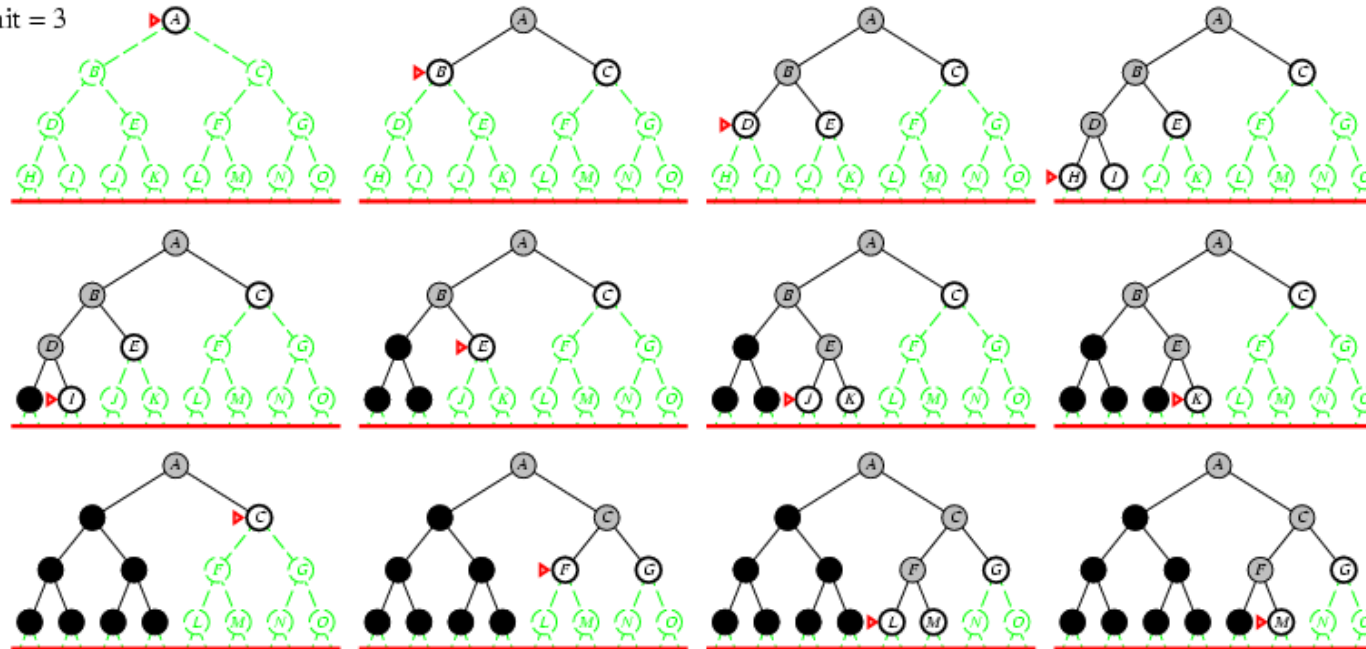
Búsqueda en Amplitud



Pontificia Universidad
JAVERIANA
Bogotá

Funcionamiento

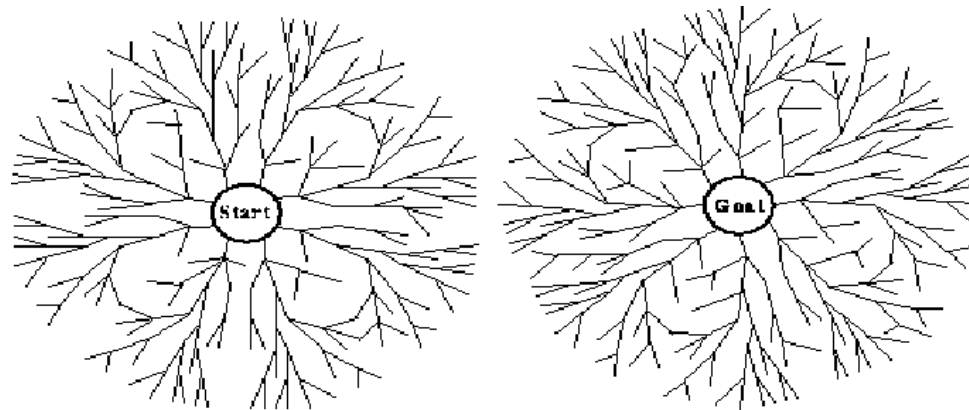
Limit = 3



Búsqueda Bidireccional

Estrategía

- Simultáneamente buscar
 - hacia adelante desde el estado inicial
 - hacia atrás desde el estado objetivo
 - se detiene cuando las dos búsquedas se encuentran en el medio



- Reducir el factor de ramificación !!

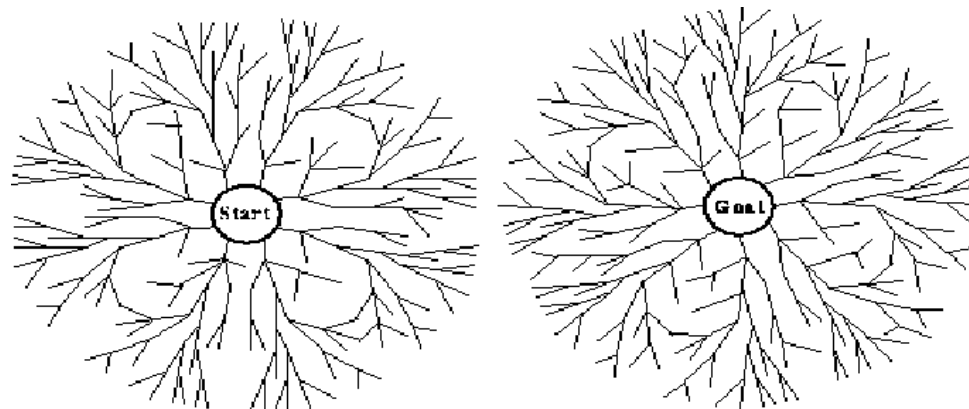
Búsqueda Bidireccional

Requerimientos Especiales

- debe definir predecesores de un nodo así como sus sucesores
- debe estar en capacidad de analizar si uno nodo generado ya fue generado en la otra búsqueda

Variantes Posibles

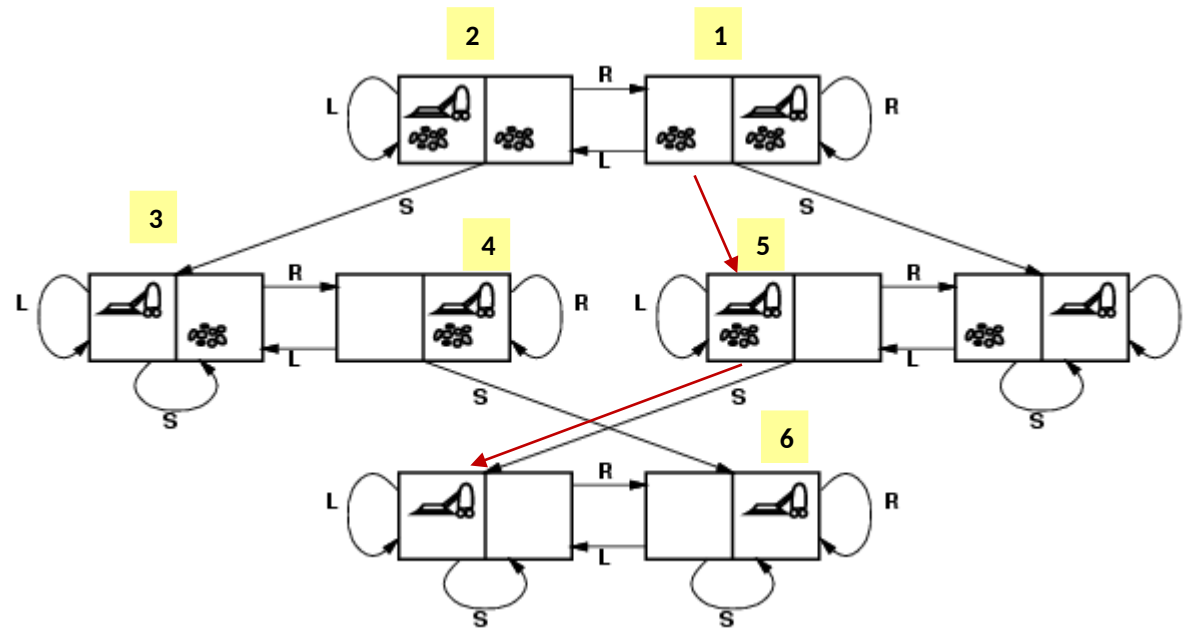
- qué pasa si hay muchos estados objetivo ?
- qué tipo de búsqueda se debe implementar en cada mitad?



Taller 2-2 – Búsqueda a Ciegas

Problema del Robot Aspiradora

A partir del análisis y modelo desarrollado en el taller 2-1, problema del robot aspiradora, realice la simulación de cómo se desarrollarían la búsqueda en profundidad y en amplitud.



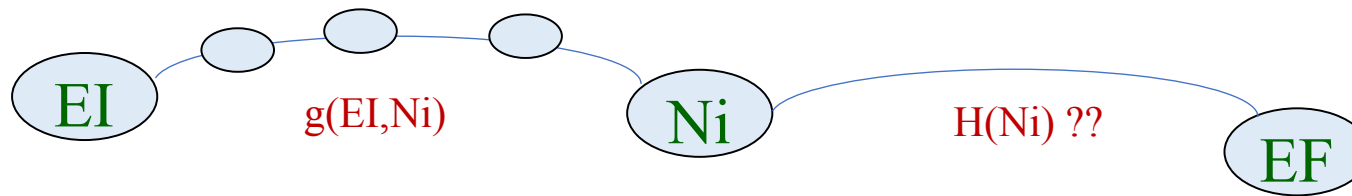
3- Búsqueda con Información

Función Heurística $\rightarrow h(N_i)$ $N_i = \text{estado } i$

- **estimar** qué tan bueno es el nodo N_i como camino candidato a alcanzar el estado final EF
- estimación basada en la experiencia/intuición
- normalmente intenta medir la “distancia más corta” desde N_i hasta EF

Función de Costo $\rightarrow g(EI, N_i)$ $EI = \text{estado inicial}$ $N_i = \text{estado } i$

- **calcular** qué tan costoso ha sido el camino para llegar a N_i desde el estado inicial EI



Función Heurística

Heurística

- conocimiento específico del dominio del problema
- ayuda a reducir el esfuerzo esperado de búsqueda

Función Heurística $\rightarrow h(N_i)$ $N_i = \text{estado } i$

- evalúan la viabilidad de expandir un nodo
- usualmente estiman costos de “distancia” desde el estado actual hasta el estado objetivo
- tiempo gastado en evaluar la heurística debería recuperarse en una reducción en tiempo de búsqueda

1	2	3
7	8	4
6		5

Initial State

	1	2
3	4	5
6	7	8

Goal

- Función Heurística 1
 - número de piezas en posición no correspondiente
- Función Heurística 2
 - distancia de Manhattan

Función Heurística - Ejemplo

Ejemplo

Planificación de Trayectorias de un Robot

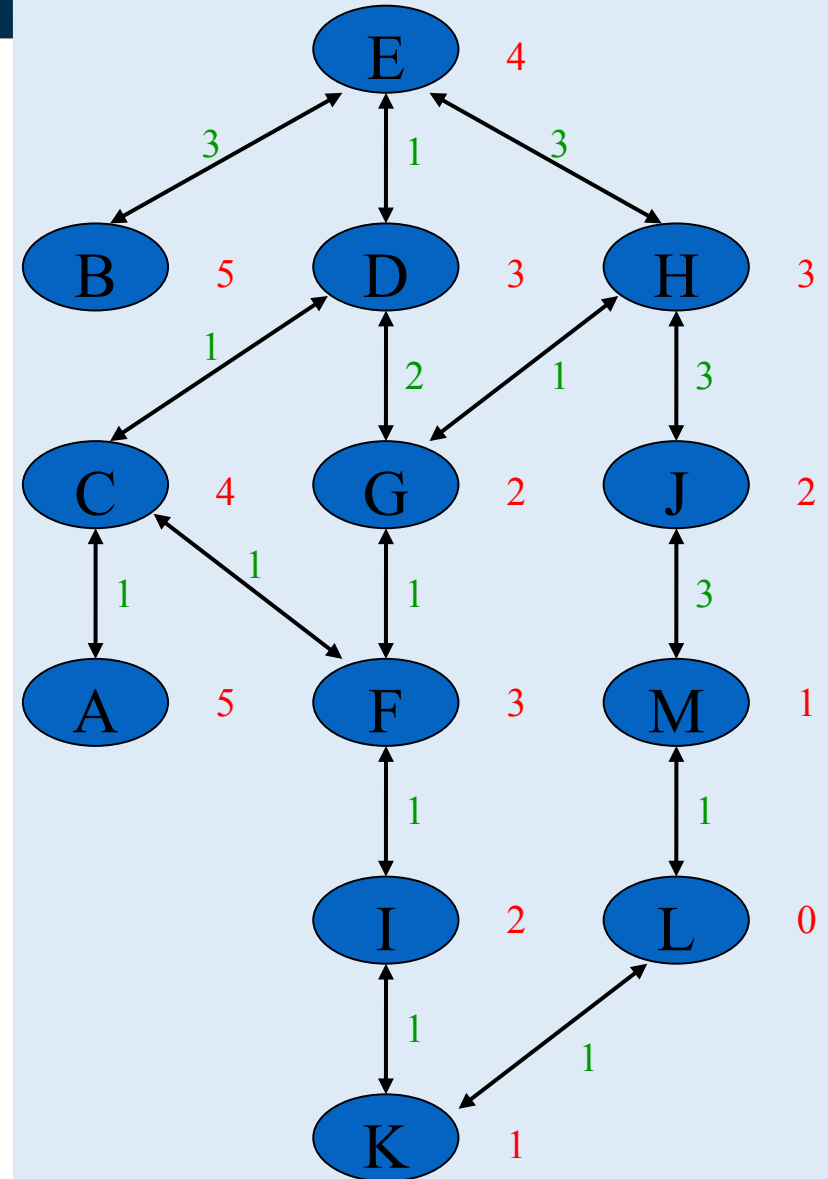
A		B
C	D	E
F	G	H
I		J
K	L	M

A		B
1		3
1	2	3
1		3
1		3
K	L	M

h Perfecta \rightarrow Pre-Calculada

h Distancia de Manhattan \rightarrow Estimada

g Costo - Horizontal 1 / Vertical 1-2-3 por líneas



Métodos de Escalada



Pontificia Universidad
JAVERIANA
Bogotá

Escalada Simple

- obtener uno de los hijos N_s y compararlo con el nodo actual N
- **si** es mejor, $h(N_s) > h(N)$ **ent** continuar con N_s **sino** se obtiene otro hijo
- **sino** hay hijos mejores **ent** terminar

Escalada con Gradiente

- comparar el nodo actual N_a con todos sus hijos N_{si}
- **si** existe hilo mejor **ent** seleccionarlo como el nuevo mejor conocido
 $h(N_m) > h(N)$ y $h(N_m) > h(N_{si}) \quad i \neq m \rightarrow N_a' = N_m$
- **sino** hay hijos mejores **ent** terminar

Problema de Mínimos Locales

- atrapado en callejones sin salida
- *backtracking* - salto aleatorio

Métodos de Escalada - Ejemplo



Pontificia Universidad
JAVERIANA

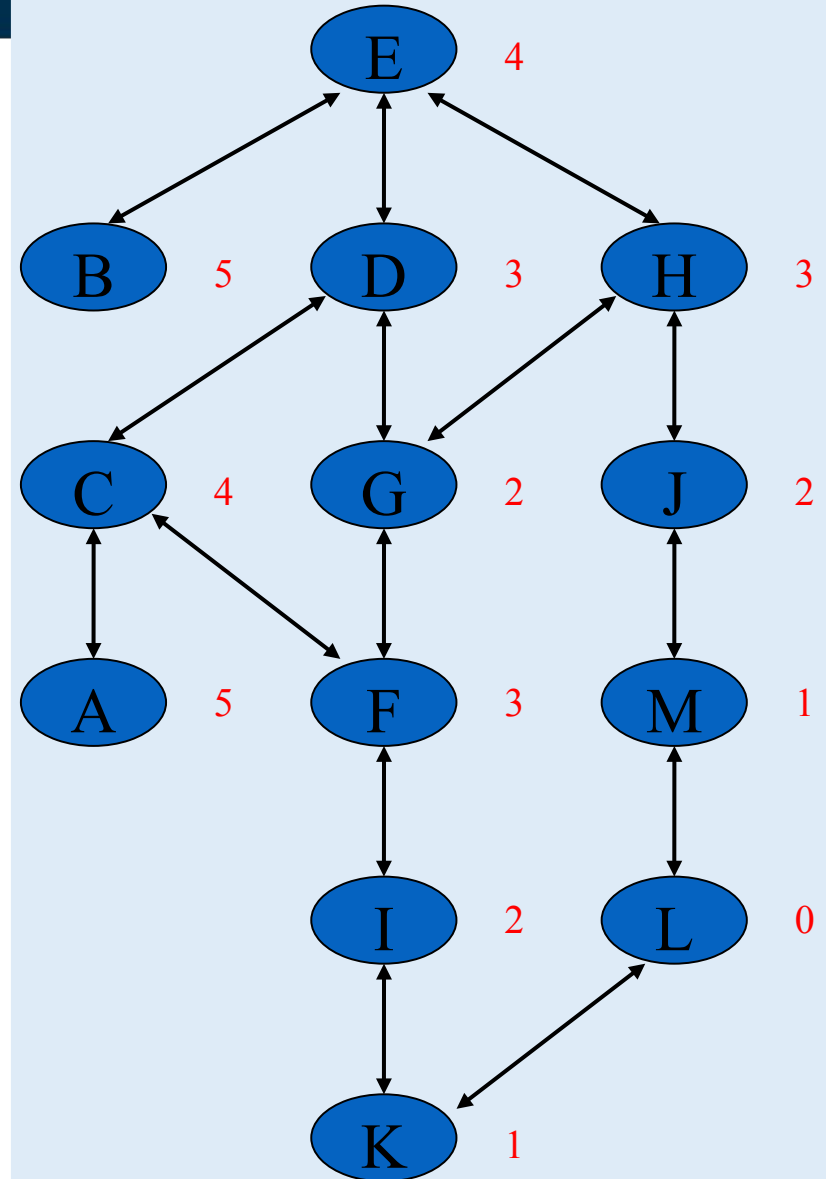
A		B
C	D	E
F	G	H
I		J
K	L	M

Escala Simple

A		B
C	D	E
F	G	H
I		J
K	L	M

Gradiente

Mínimos Locales



Best First



Pontificia Universidad
JAVERIANA
Bogotá

Best First

- generar todos los hijos N_{si} del nodo actual N_a
- agregar los N_{si} a la lista de los nodos abiertos por explorar
- seleccionar como nuevo nodo a explorar $N_{a'}$ al de mejor $h(N_m)$ de todos los nodos abiertos aún no explorados
 $h(N_m) > h(N_j) \quad j \neq m$
marcar N_a como explorado $\rightarrow N_{a'} = N_m$
si no hay más nodos por explorar **ent** no hay solución

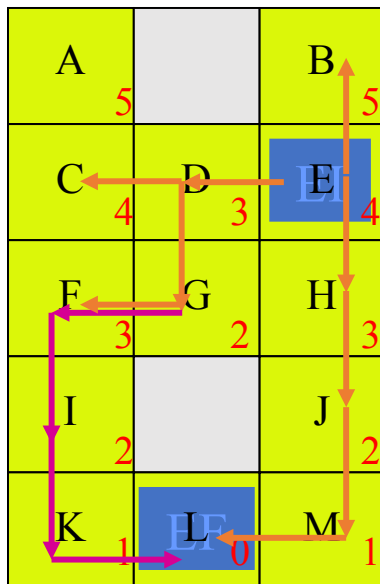
Características

- No siempre es óptimo
 - pero si es guiado por heurística
- Completo
- Requiere mucha memoria \rightarrow múltiples listas
- No hay mínimos locales

Best First - Ejemplo

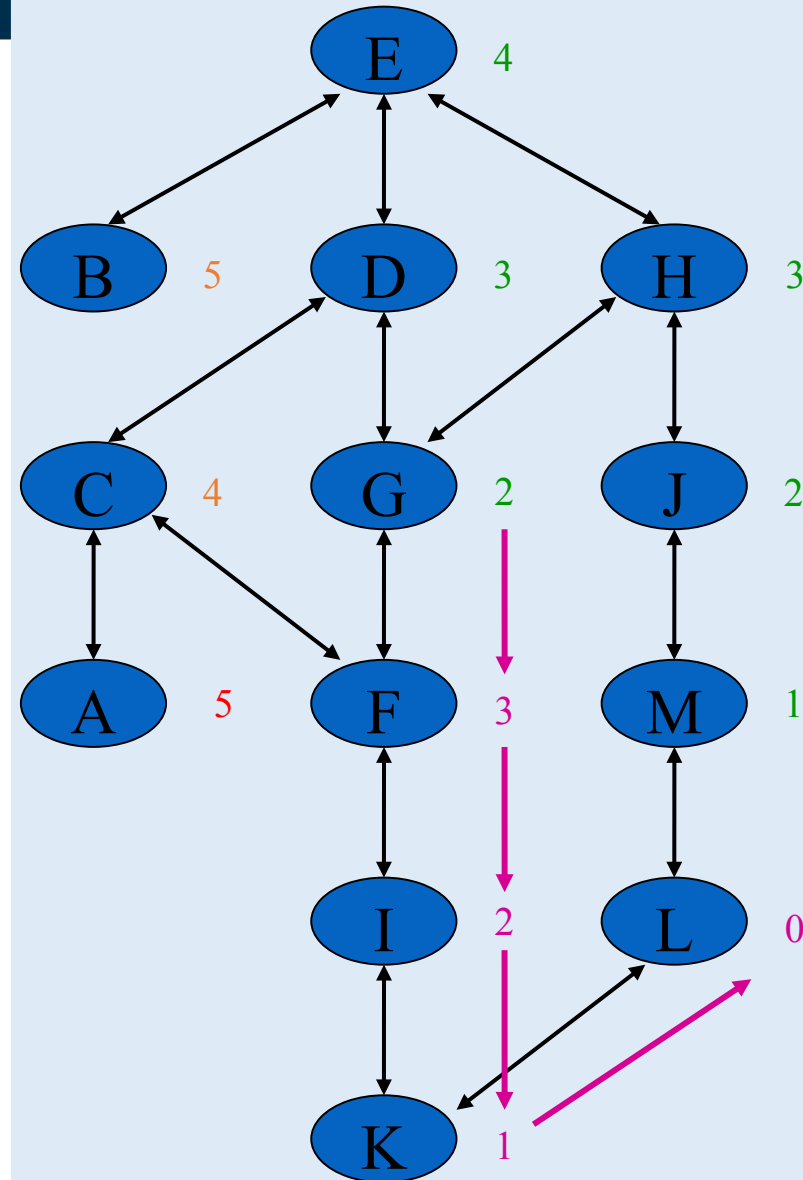


Pontificia Universidad
JAVERIANA



Heurística Guía
la Búsqueda

No Siempre
es Optimo !!



Costo Uniforme



Pontificia Universidad
JAVERIANA
Bogotá

Costo Uniforme

- generar todos los hijos N_{si} del nodo actual N_a
- agregar los N_{si} a la lista de los nodos abiertos por explorar
- seleccionar como nuevo nodo a explorar $N_{a'}$ al de menor costo de todos los nodos abiertos aún no explorados
 $g(N_m) > g(N_j) \quad j \neq m$
marcar N_a como explorado $\rightarrow N_{a'} = N_m$
si no hay más nodos por explorar **ent** no hay solución

Características

- Sí es óptimo
 - pero no es guiado por heurística
- Completo
- Requiere mucha memoria \rightarrow múltiples listas
- No hay mínimos locales

Algoritmo A*



Pontificia Universidad
JAVERIANA
Bogotá

A*

- similar a Best First y Costo Uniforme
- función de selección compuesta

$$f(N) = g(N) + h(N)$$

$g(N)$: costo del mejor camino

$h(N)$: estimación del costo del camino hacia la solución

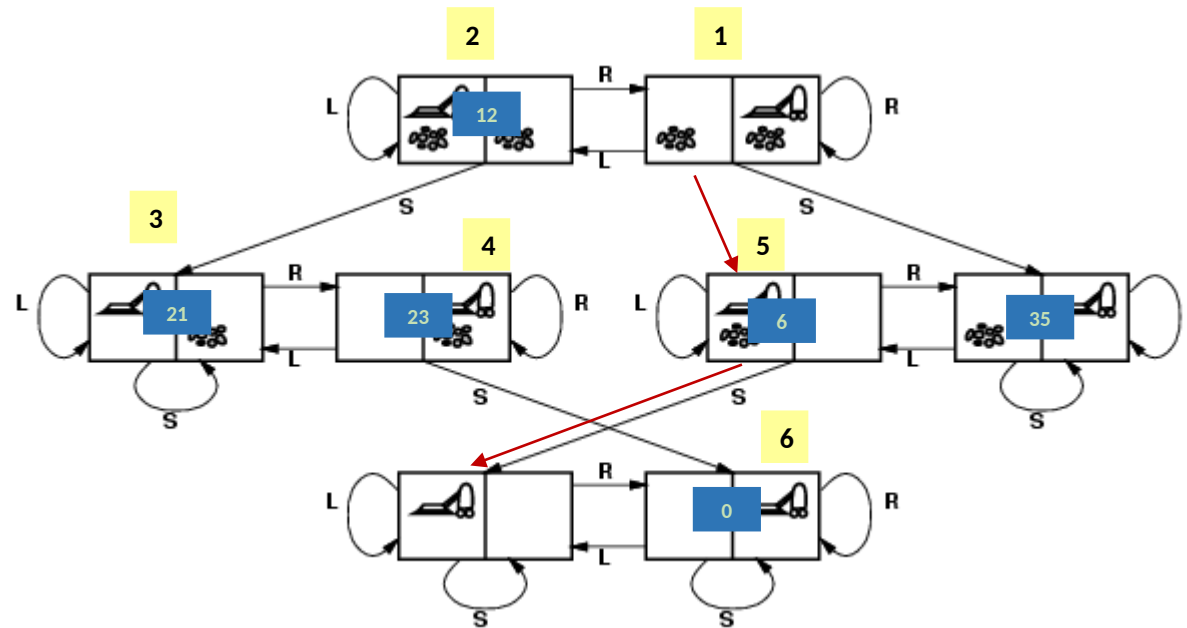
Características

- Sí es guiado por heurística y Sí es óptimo
 - Garantizado OK a condición de que $h(N_i)$ sea “**optimista**”
- Completo
- Requiere mucha memoria → múltiples listas
- No hay mínimos locales

Taller 2-3 – Búsqueda Informada

Problema del Robot Aspiradora

A partir del análisis y modelo desarrollado en el taller 2-1, problema del robot aspiradora, realice la simulación de cómo se desarrollaría la búsqueda Best First.



Bibliografía



- 1. Rusell N., Inteligencia Artificial: Un Enfoque Moderno, Prentice Hall, 2004.
- 2. Todos los libros de IA clásica incluyen este tema.
 - Rich N., Inteligencia Artificial, McGraw Hill, 1998.
 - Nilsson N.J., Inteligencia artificial: Una Nueva Síntesis, McGraw Hill, 2001.
 - Korf R.E, Artificial Intelligence Search Algorithms, 1996.
 - Hansen E. Notas de Clase, Artificial Intelligence Mississippi State University, Spring 2002.



Pontificia Universidad
JAVERIANA
Bogotá

Inteligencia Artificial

Búsquedas en Espacio del Problema

Ing. Andrea Rueda, PhD – rueda-andrea@javeriana.edu.co

Ing. Enrique González, PhD – egonzal@javeriana.edu.co

Departamento de Ingeniería de Sistemas