

# **Fundamentos de Inteligencia Artificial**

Centro Asociado de Melilla

Tutor: Aziz Mulud Tieb

# Bloque 3:

## Técnicas Basadas en Búsquedas Heurísticas

**3.1 Concepto de búsqueda heurística**

**3.2 Primero el mejor**

**3.3 Algoritmo A\***

## 3.1- Concepto de búsqueda heurística

- Las técnicas de búsqueda heurística usan el conocimiento del dominio para adaptar el solucionador y, de esta manera, éste sea más potente y consiga llegar a la solución con mayor rapidez.
- Reducir anchura del árbol expandido: **eliminar algunas ramas** porque probablemente no reflejan la solución óptima
- Estas técnicas utilizan el conocimiento para avanzar buscando la solución al problema.
- **Definiciones:**
  - **Costo del camino:** coste necesario para ir del nodo raíz al nodo meta por dicho camino.
  - **Costo para hallar la solución:** coste necesario para encontrar el camino anteriormente definido.
- **Potencia heurística:** capacidad de un método de exploración para obtener la solución con un coste lo más bajo posible.
- La principal diferencia de este tipo de búsqueda es que ahora a cada nodo se le va a poder asociar un valor que dará idea de lo cerca que se encuentra de un nodo meta.

# Función de evaluación heurística

- **Definición:** es una aplicación del espacio de estados con el espacio de los números reales:

$$F(\text{estado}) = n$$

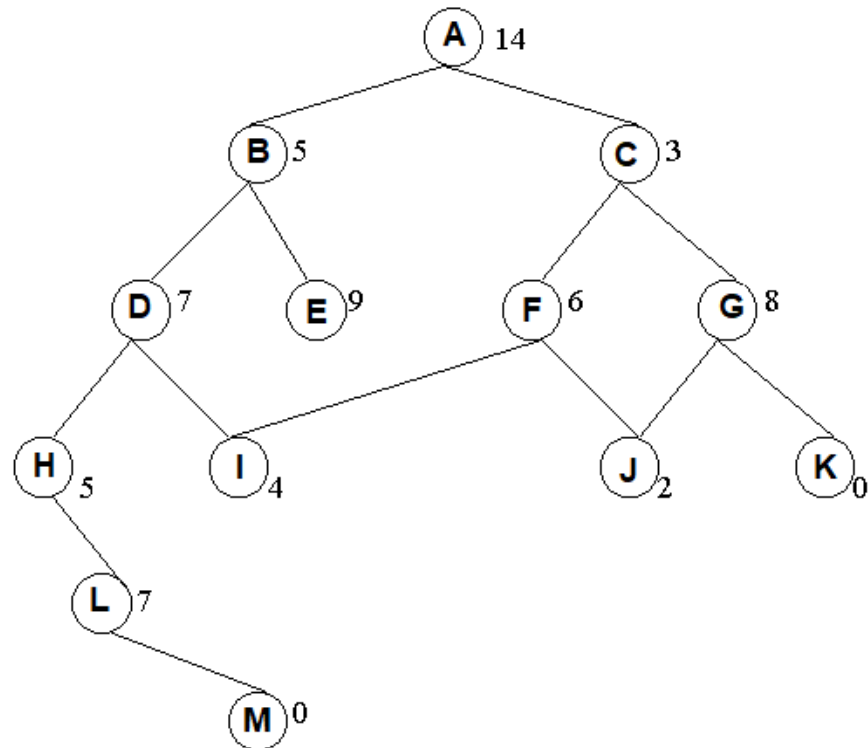
- $n$  representa lo cercano que esta el estado con el que se ha aplicado la función de evaluación de la solución final.
- Es muy importante mantener un equilibrio entre la eficiencia de la función y su complejidad.
- La función de evaluación heurística (fev) no es más que una **estimación de la distancia real a la meta**.
- La fev permitirá guiar la búsqueda hacia aquellos camino que se supone son más prometedores.

## 3.2 Primero el mejor

- ❑ Los nodos se ordenan de tal manera que se expande el nodo de mejor valor de la función heurística ***f***.
- ❑ ***f(n)*** da un valor numérico que indica una medida de lo prometedor que es el nodo para ser expandido.
- ❑ Esta función se utiliza para ***ordenar*** la lista ***ABIERTA***, de modo que aquellos nodos más prometedores estarán al principio.
- ❑ Esta función puede tener en cuenta o no el costo del mejor camino parcial encontrado en cada momento desde la raíz hasta el nodo considerado.
- ❑ Complejidad temporal  $O(b^p)$ . *b*/factor de ramificación. *p*/ profundidad máx
- ❑ Complejidad espacial:  $O(b^p)$ . Depende del problema concreto
- ❑ ***No es completa***, en general.

Por ejemplo, una mala heurística podría hacer que se tomara un camino infinito.
- ❑ No es ***minimal*** ya que no garantiza soluciones con el menor número de operadores.
- ❑ La heurística podría guiar hacia una solución no minimal

## 3.2 Primero el mejor: Ejemplo



ABIERTA

Paso 1) A (14)

Paso 2) C(3), B(5)

Paso 3) B(5), F(6), G(8)

Paso 4) F(6), D(7), G(8), E(9)

Paso 5) J(2), I(4), D(7), G(8), E(9)

Paso 6) I(4), D(7), G(8), E(9)

TABLA\_A

Vacía

A

A,C,

A,C,B,

A,C,B,F

A,C,B,F,J

En este ciclo del algoritmo se encuentra un descendiente (G) del nodo que se está expandiendo (J), ya que estaba en ABIERTA. A pesar de ello no se produce reorientación por ser el anterior camino desde G al nodo raíz de menor costo que el encontrado ahora.

Paso 7) D(7), G(8), E(9)

A,C,B,F,J,I,

Tenemos un caso análogo al del ciclo anterior.

Paso 8) H(5), G(8), E(9)

A,C,B,F,J,I,D

Al expandir D se genera I, que ya estaba TABLA\_A. El nuevo camino encontrado hasta I no es menos costoso que el anterior; por lo tanto, no hay que redirigir arcos.

Paso 9) L(7), G(8), E(9)

A,C,B,F,J,I,D,H

Paso 10) G(8),E(9)

A,C,B,F,J,I,D,H,L

Con la generación de M se llega al final del algoritmo. La lista ABIERTA debe mantenerse ordenada a lo largo de todo el proceso, ya que el "siguiente nodo a expandir" es el primero de dicha lista.

El camino encontrado siguiendo los punteros, que siempre deben señalar al mejor antecesor de un nodo en el grafo, es: A, B, D, H, L, M

## 3.3 Algoritmo A\*

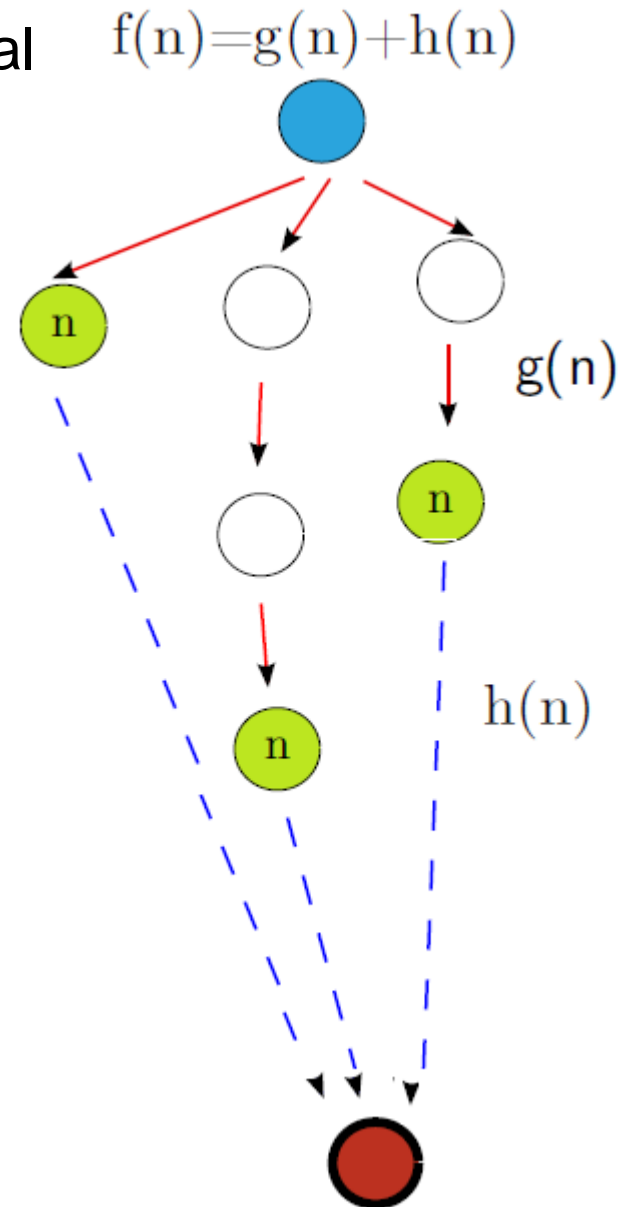
La función de evaluación tiene dos componentes:

1. coste para ir desde el (un) inicio al nodo actual
2. coste (estimado) para ir desde el nodo actual a una solución

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

- ☐  $f$  es un valor **estimado** del coste total del camino que pasa por  $n$
- ☐  $h$  (heurístico) es un valor **estimado** de lo que falta para llegar desde  $n$  al (a un) objetivo
- ☐  $g$  es un coste **real** (lo gastado por el camino más corto **conocido** hasta  $n$ )

La **preferencia** es siempre del nodo con menor  $f$ , en caso de empate, la preferencia es del nodo con menor  $h$ .



## 3.3 Algoritmo A\*: código

---

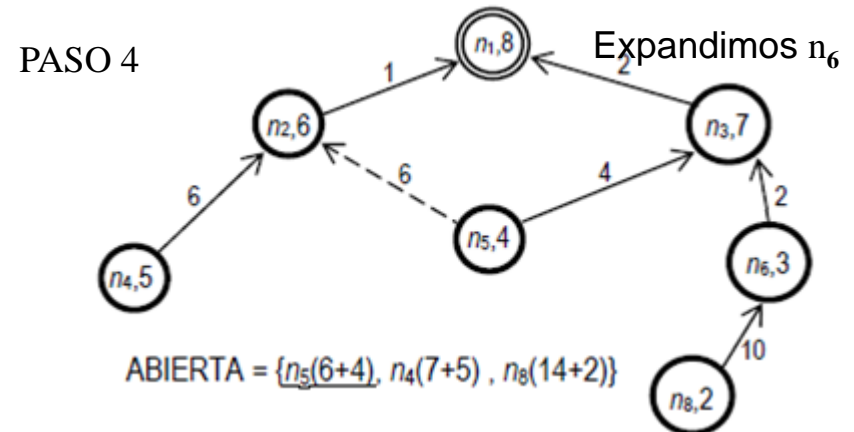
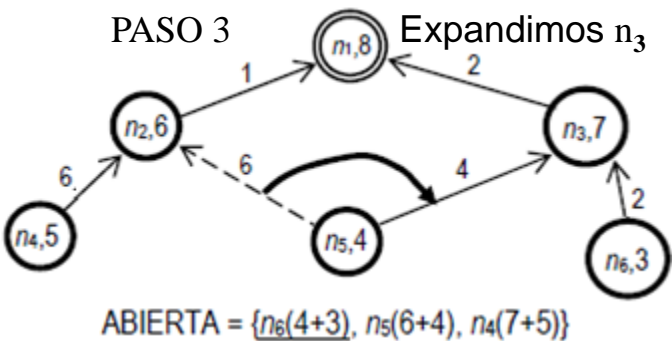
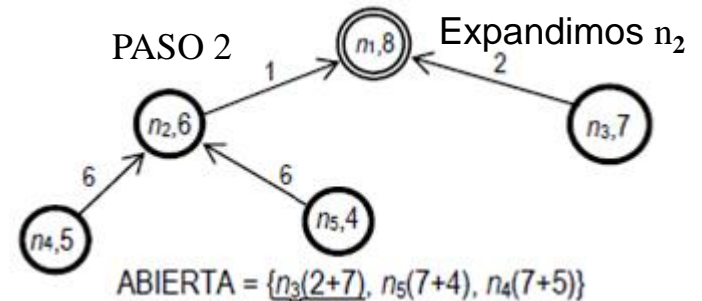
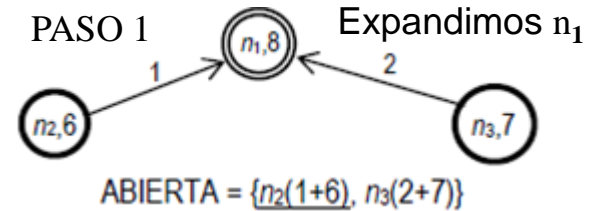
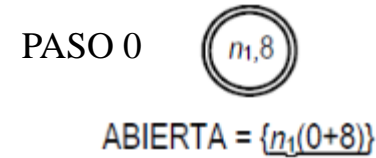
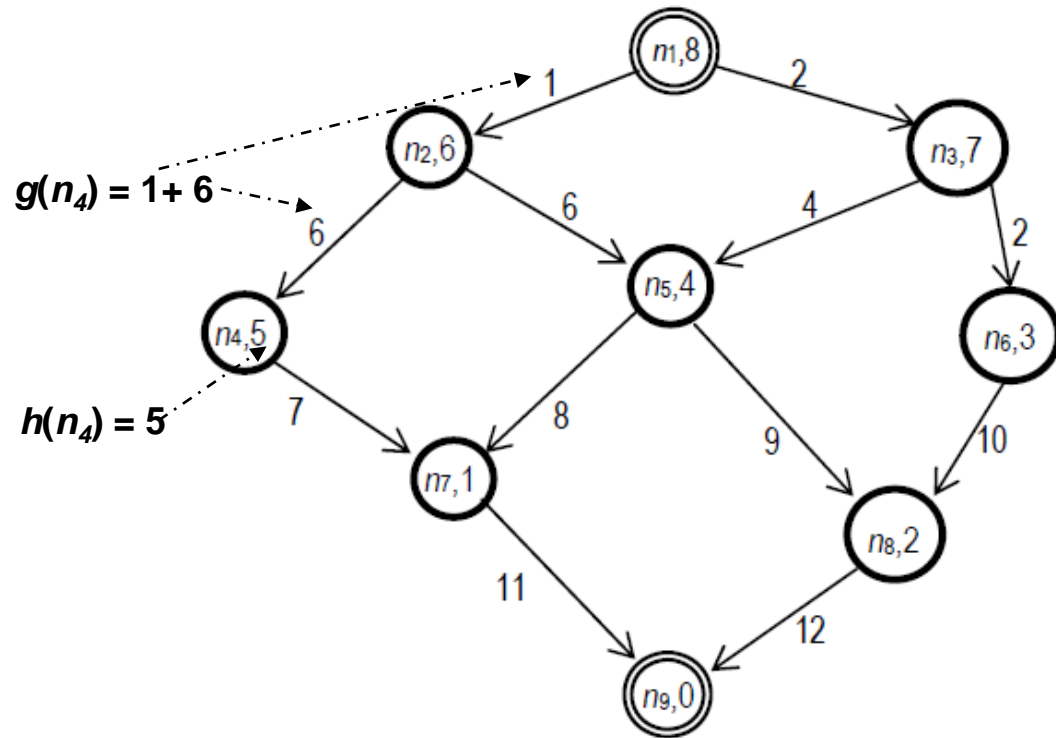
```
ABIERTA = (inicial);
mientras NoVacía(ABIERTA) hacer
     $n = \text{ExtraePrimero}(\text{ABIERTA});$ 
    si EsObjetivo( $n$ ) entonces
        devolver Camino( $\text{inicial}, n$ ) y para;
    fin si
     $S = \text{Sucesores}(n);$ 
    Añade  $S$  a la entrada de  $n$  en la TABLA_A;
    para cada  $q$  de  $S$  hacer
        si ( $q \in \text{TABLA\_A}$ ) entonces
            Rectificar( $q, n, \text{Coste}(n, q)$ );
            Ordenar(ABIERTA); {si es preciso}
        si no
            pone  $q$  en la TABLA_A con
                Anterior( $q$ ) =  $n$ ,
                 $g(q) = g(n) + \text{Coste}(n, q)$ ,
                 $h(q) = \text{Heuristico}(q)$ ;
            ABIERTA = Mezclar( $q, \text{ABIERTA}$ );
        fin si
    fin para
fin mientras
devolver “no solución”;
```

---



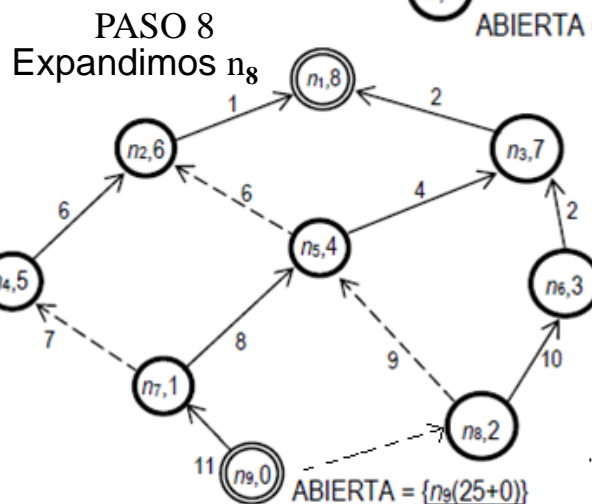
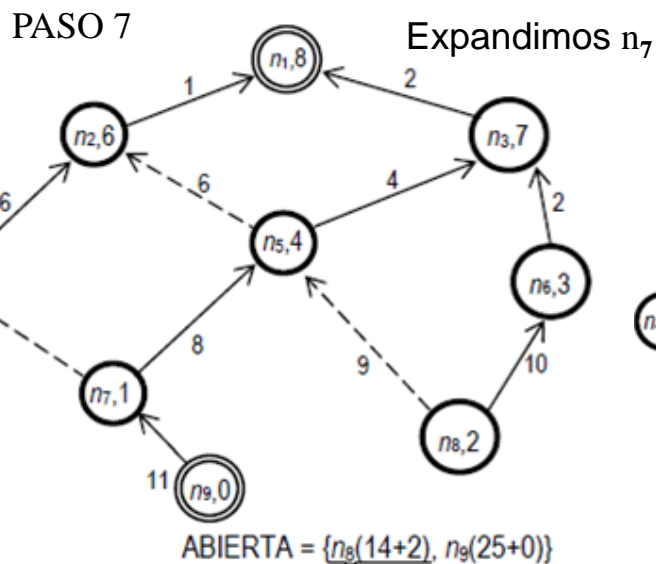
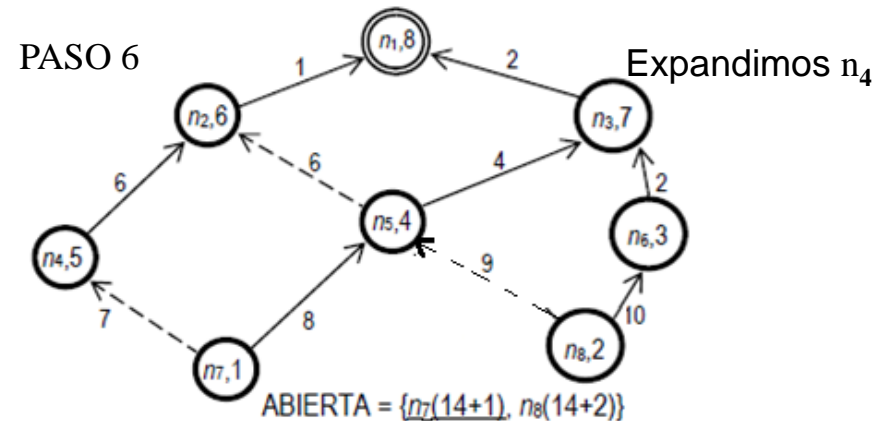
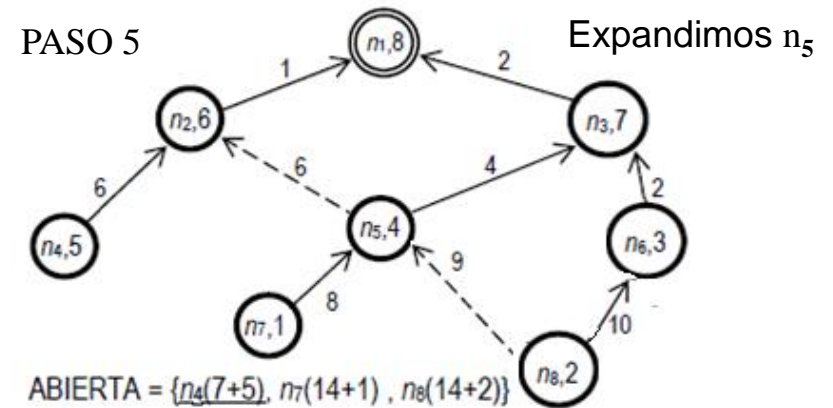
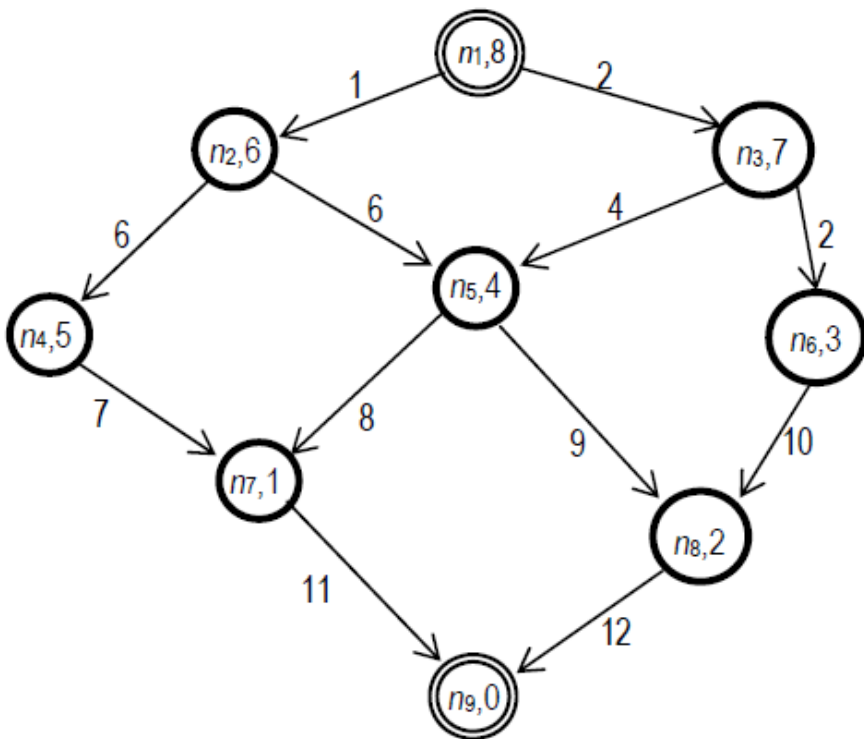
# 3.3 Algoritmo A\*: Ejemplo

$$f(n_4) = g(n_4) + h(n_4)$$



El mejor camino desde  $n_5$  al nodo inicial lo marca su padre  $n_3$  (coste  $4+2=6$ ) y no  $n_2$  (coste  $6+1=7$ ). El arco de  $n_5$  a  $n_3$  se marca con trazo **continuo** y de  $n_5$  a  $n_2$  con trazo **discontinuo**

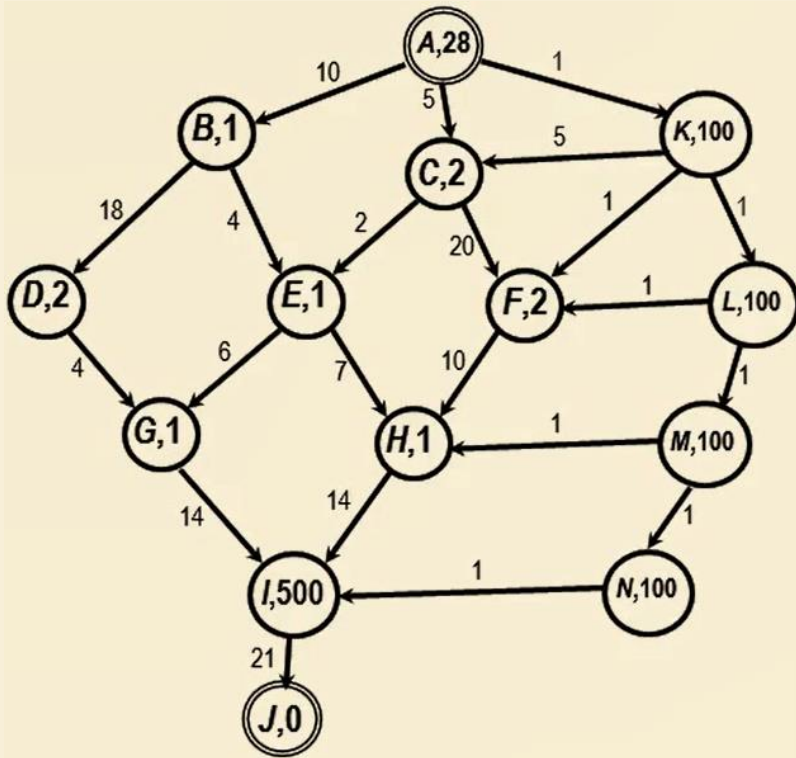
# 3.3 Algoritmo A\*: Ejemplo



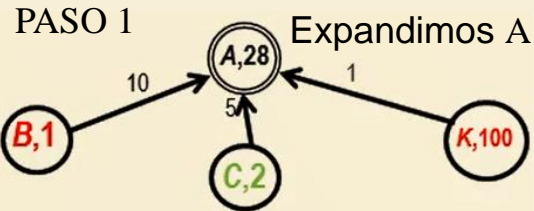
## PASO 9

Intentamos expandir  $n_9$  y alcanzamos una meta, con lo que el algoritmo termina. El camino solución encontrado es:  $n_1 \rightarrow n_3 \rightarrow n_5 \rightarrow n_7 \rightarrow n_9$ , cuyo **coste es 25**.

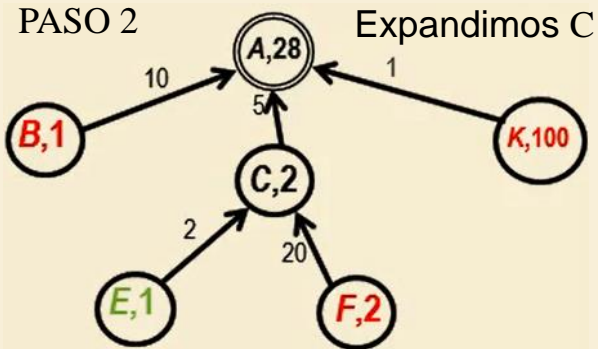
# 3.3 Algoritmo A\*: Ejemplo2



PASO 0 **A,28** ABIERTA = {A(0+28)}

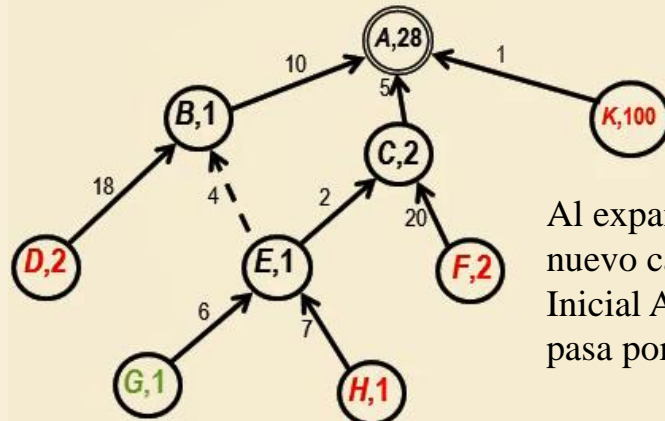


ABIERTA = {C(5+2), B(10+1), K(1+100)}



ABIERTA = {E(7+1), B(10+1), F(25+2), K(1+100)}

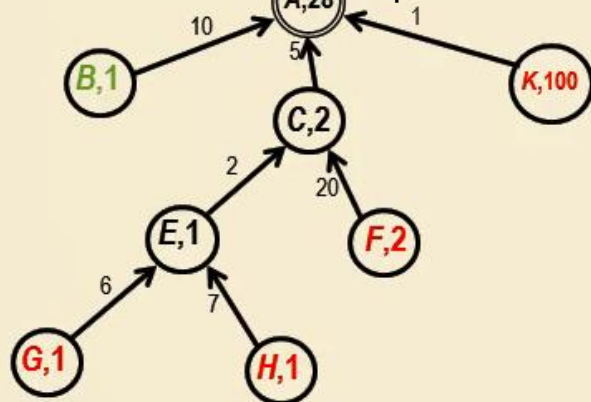
PASO 4 Expandimos B



ABIERTA = {G(13+1), H(14+1), F(25+2), D(28+2), K(1+100)}

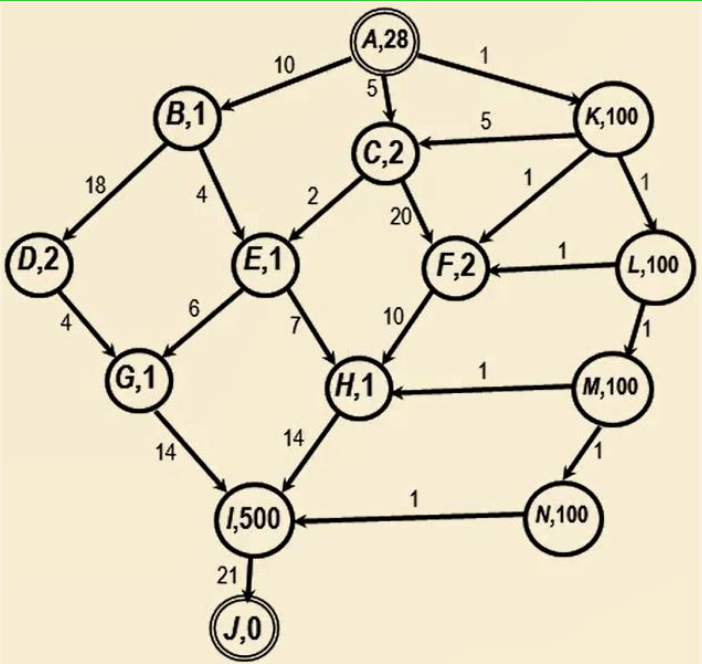
Al expandir B, encontramos un nuevo camino del nodo E al nodo Inicial A. El mejor camino de E a A pasa por C y no por B.

PASO 3 Expandimos E



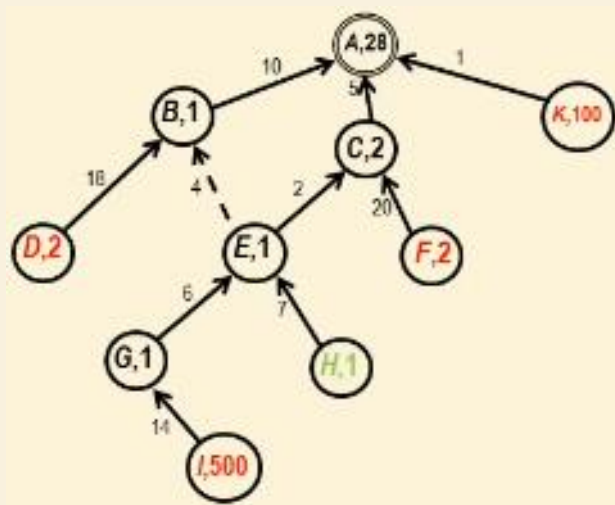
ABIERTA = {B(10+1), G(13+1), H(14+1), F(25+2), K(1+100)}

# 3.3 Algoritmo A\*: Ejemplo2



PASO 5

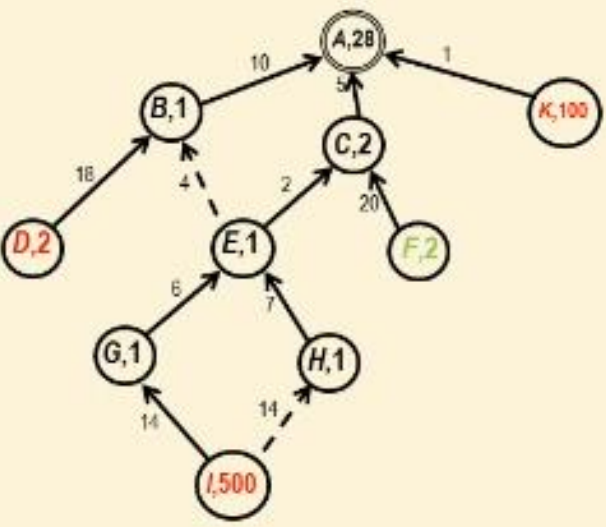
Expandimos G



ABIERTA = {H(14+1), F(25+2), D(28+2), K(1+100), I(27+500)}

PASO 6

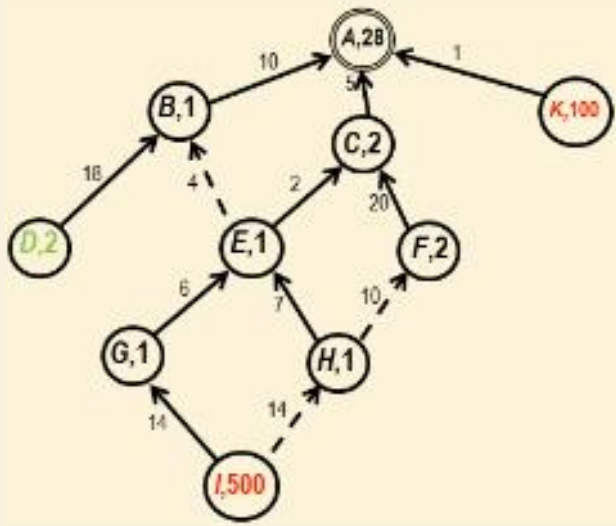
Expandimos H



ABIERTA = {F(25+2), D(28+2), K(1+100), I(27+500)}

PASO 7

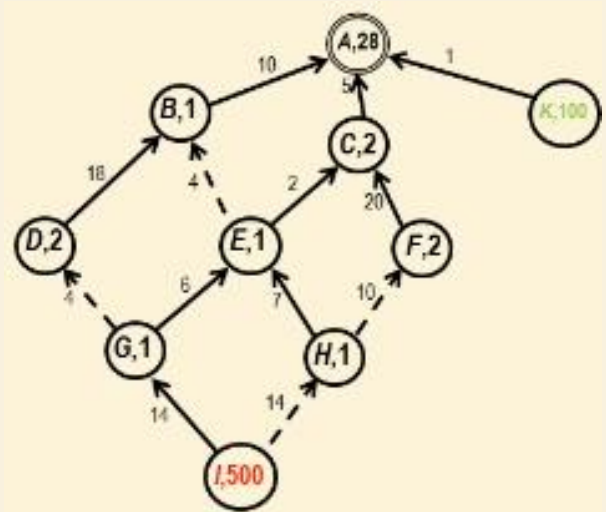
Expandimos F



ABIERTA = {D(28+2), K(1+100), I(27+500)}

PASO 8

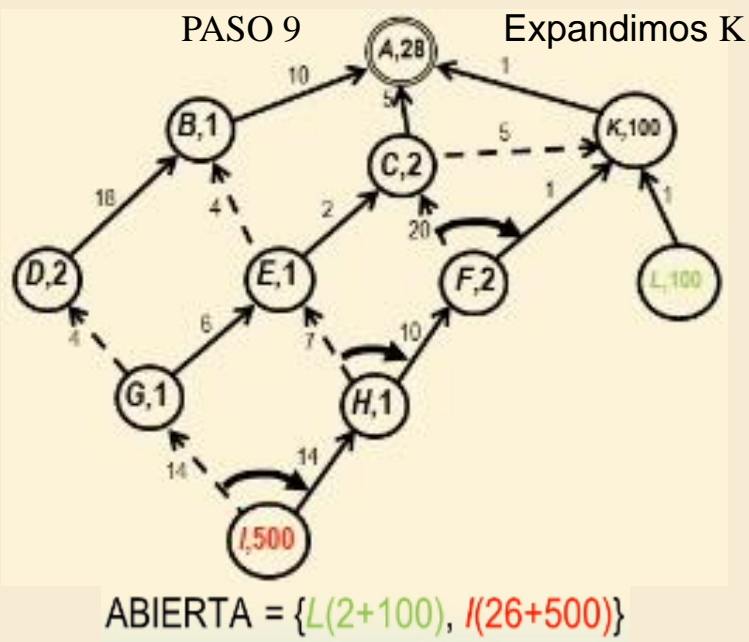
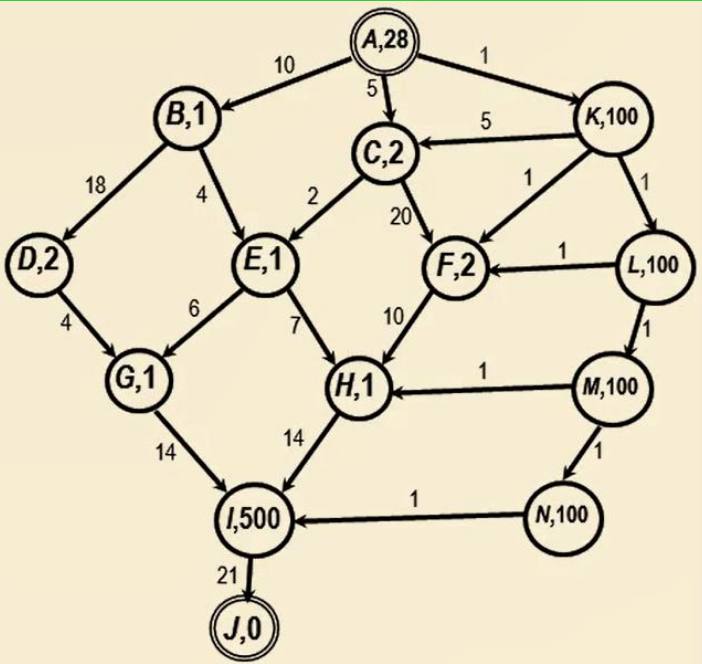
Expandimos D



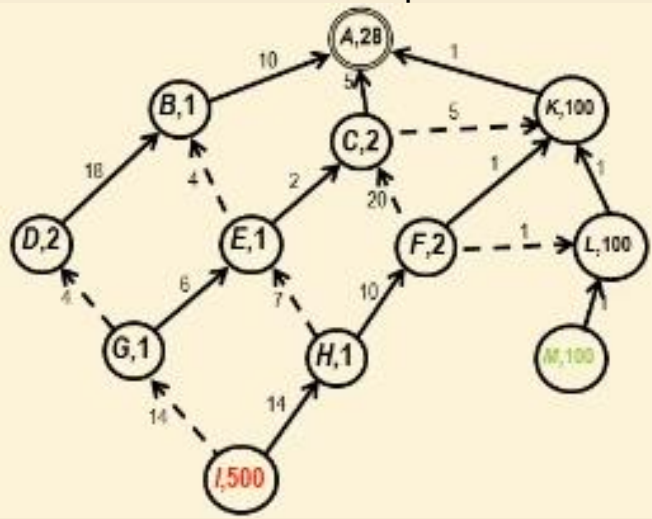
ABIERTA = {K(1+100), I(27+500)}



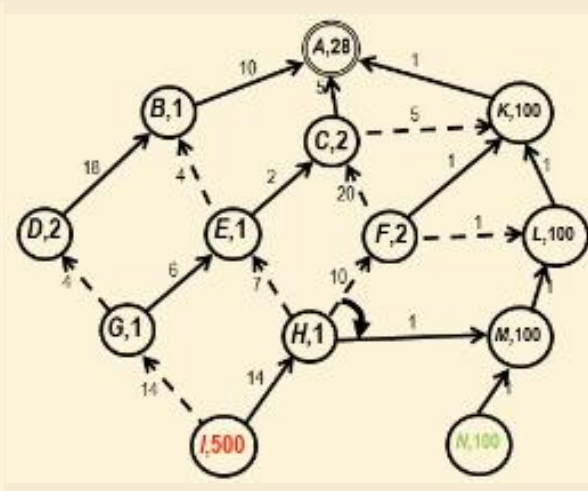
# 3.3 Algoritmo A\*: Ejemplo2



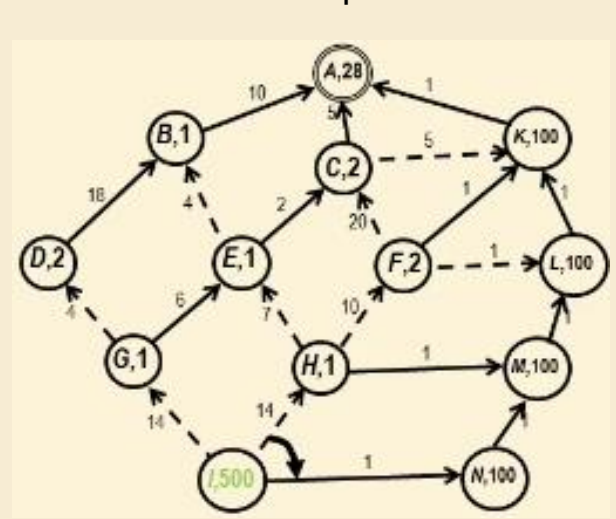
PASO 10 Expandimos L



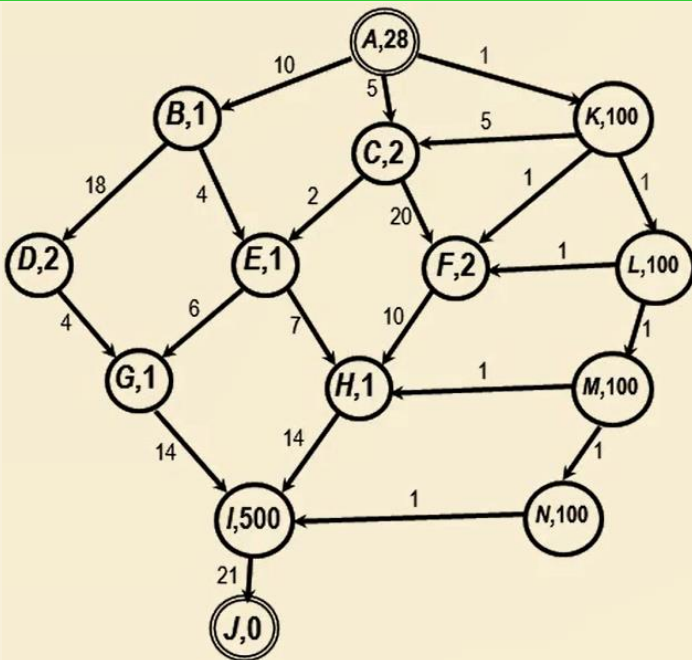
PASO 11 Expandimos M



PASO 12 Expandimos N

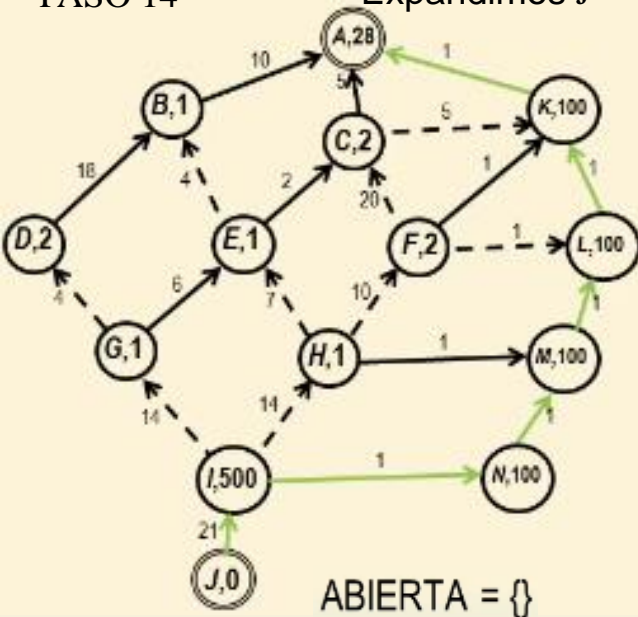


# 3.3 Algoritmo A\*: Ejemplo2



PASO 14

Expandimos J



PASO 11

Expandimos M

PASO 12

Expandimos N

Camino solución → **A K L M N J**  
 Coste → **26**

PASO 13

Expandimos I

