

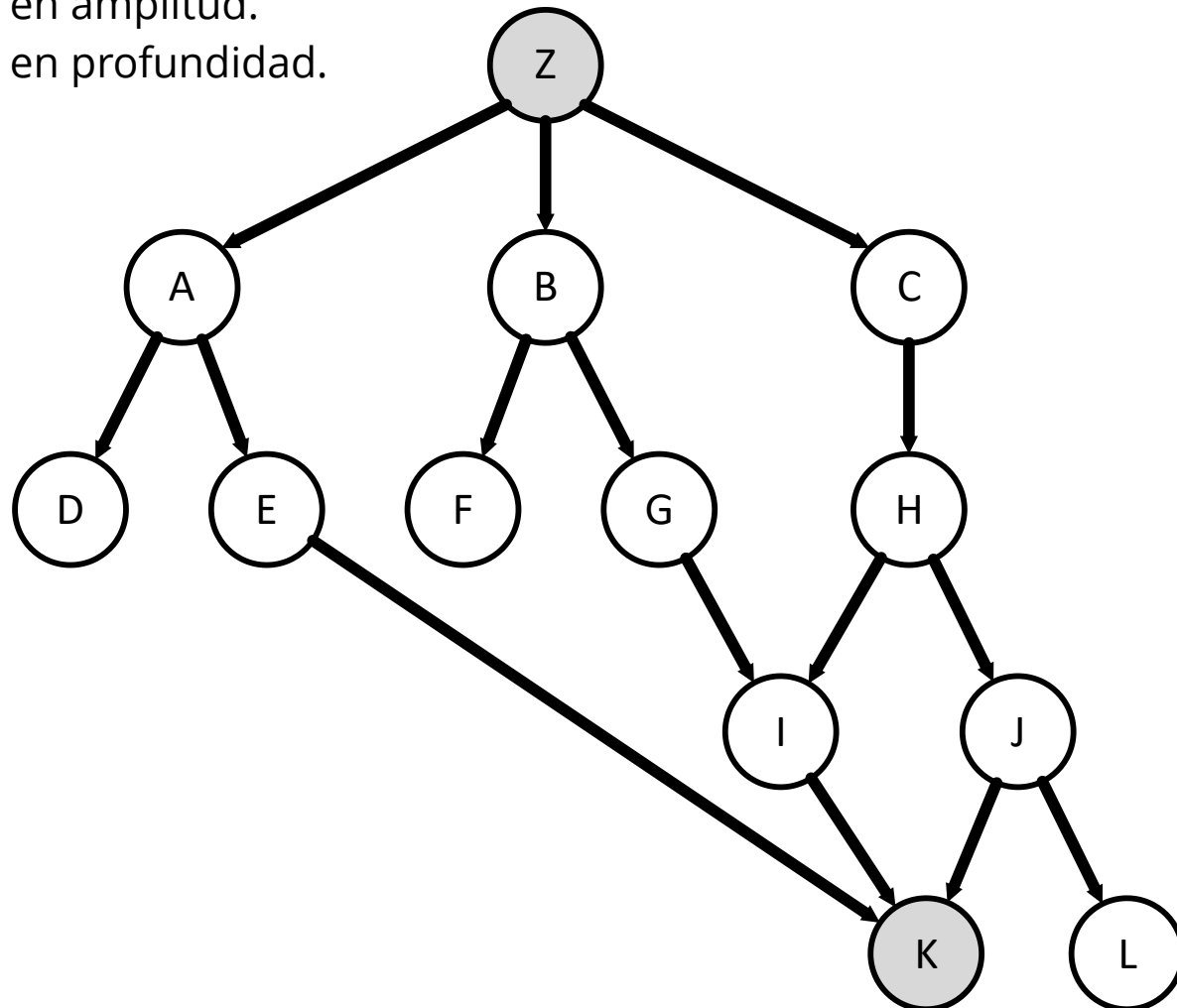
# Ejercicios Búsqueda no informada

## Amplitud, profundidad

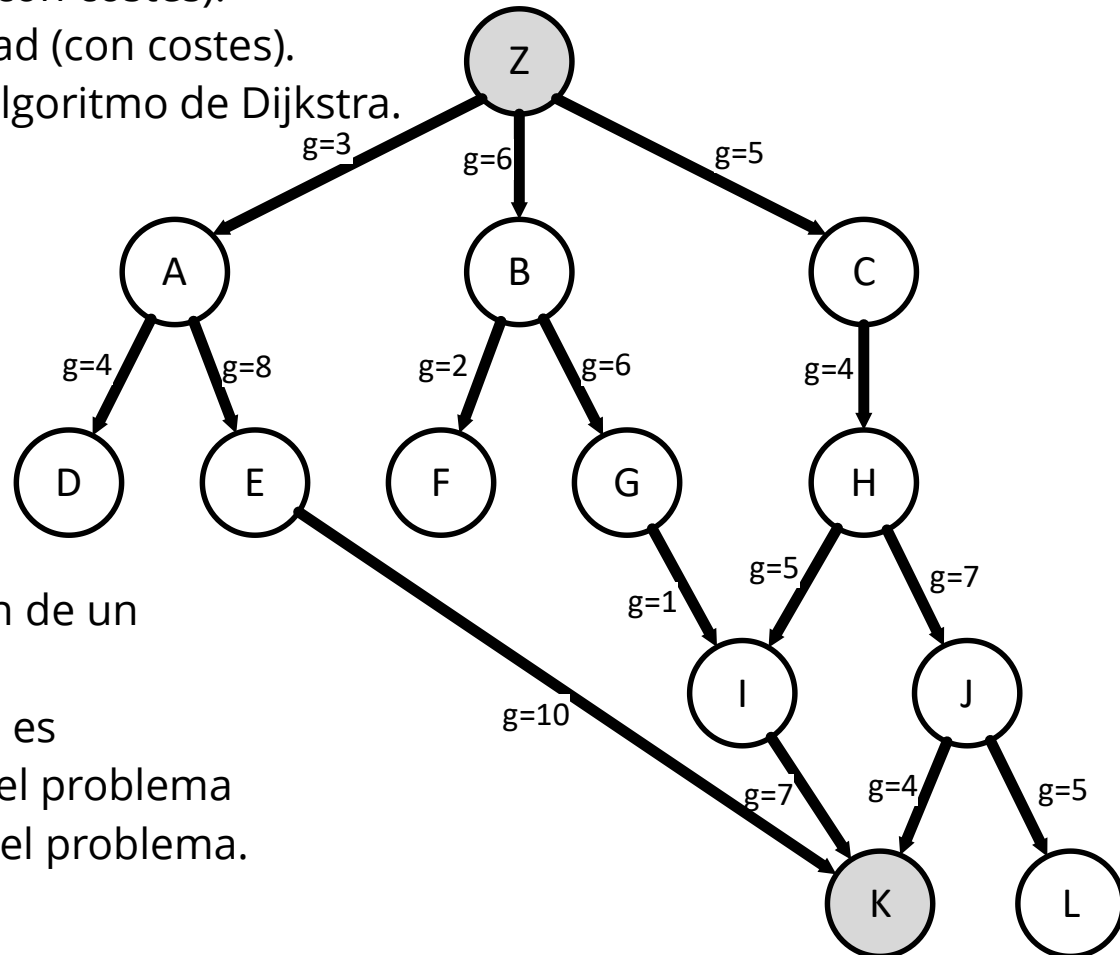
---

Inteligencia Artificial Colmenarejo  
Curso 2022-2023

- Resolución de un grafo **sin costes** mediante búsqueda no informada.
  - Búsqueda en amplitud.
  - Búsqueda en profundidad.



- Resolución de un grafo **con costes** mediante búsqueda no informada.
  - Búsqueda en amplitud (con costes).
  - Búsqueda en profundidad (con costes).
  - Búsqueda mediante el algoritmo de Dijkstra.



- Procedimiento** de resolución de un problema de búsqueda.
  - Justificar** qué algoritmo es necesario para resolver el problema según lo que se pida en el problema.
  - Y **resolverlo**.

- **Representación** de un problema de búsqueda.  
Se os proporciona un enunciado y se tiene que detallar la siguiente información para una solución mediante búsqueda no informada.
  1. Definir **información del estado**.
    - Es decir, qué información almacena cada nodo del grafo para ser único en el problema respecto a los demás.
  2. Definir **estados inicial y final**.
    - Nodo inicial y final del grafo.
  3. Definir **acciones** u operadores (genéricas).
    - Es decir, las aristas entre nodos, dirigidas o no. Qué se puede hacer para pasar del estado (nodo) actual a un nodo hijo.

Esto se puede hacer usando la misma sintaxis de un sistema de producción: hechos para los estados, y reglas para los operadores (ver introducción de los ejercicios de inferencia).

4. Especificar el **espacio de búsqueda** del problema (si se pide).

- **Representación** de un problema de búsqueda.
- 4. Especificar el **espacio de búsqueda** del problema (si se pide).
  - Mediante combinatoria indicar el valor (o su fórmula si no se puede calcular) de estados posibles a tomar en el problema. Para ello hay que analizar todas las posibilidades existentes: acciones, posiciones, etc. para el problema dado.

### Consideraciones:

- Para definir un estado hay que escribir toda la información que lo caracteriza.
  - En los ejercicios de grafos todo el estado se representa con una sola letra (A, B) junto a su coste, si lo hubiera.
  - En otros ejercicios es posible necesitar estructuras como tuplas, listas, u objetos para almacenar TODA la información que identifique a dicho instante dentro del problema.
- El formato de los estados y acciones tiene que escribirse de forma genérica (con variables).
- El estado inicial o final se escriben con valores particulares para estas variables.

## Aplicación de algoritmos de Búsqueda

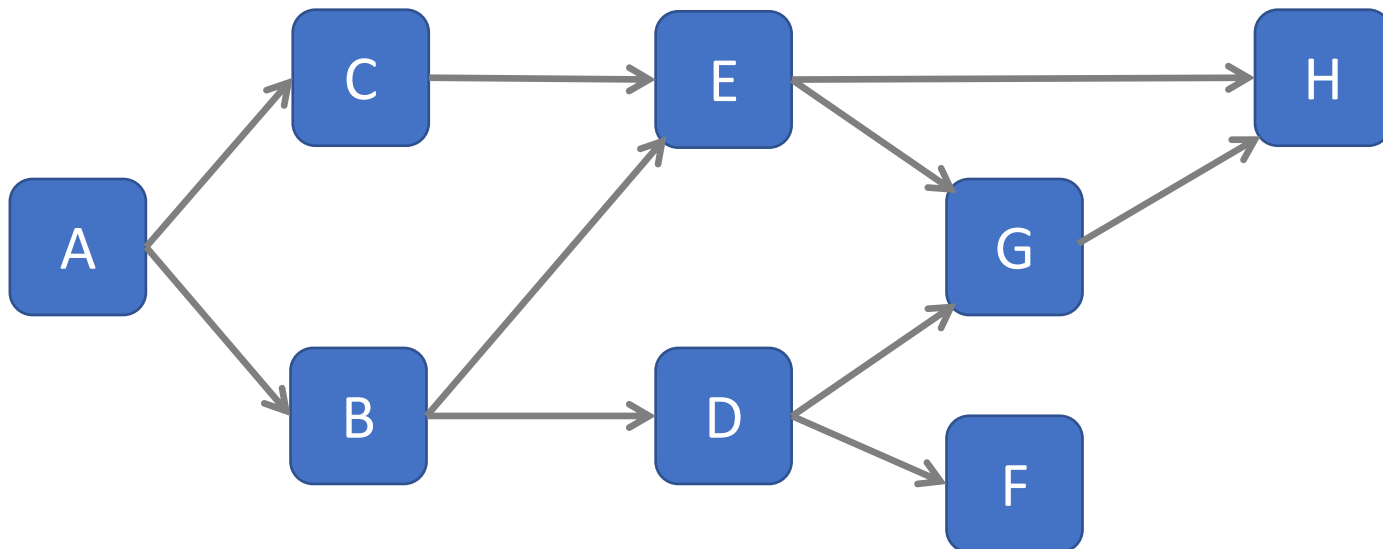
- Hay que simular la ejecución de un algoritmo no basta indicar la solución mientras se deja claro el orden de expansión de los nodos. Se puede usar cualquier formato mientras el profesor lo comprenda. Por ejemplo, se proponen los siguientes (y se ven en los ej.):
  - Una tabla en la que cada línea muestre la acción tomada, nodo expandido, y el estado del algoritmo (lista abierta con todos los detalles necesarios).
  - Un árbol donde cada nodo contiene un estado, etiquetado con el orden en que ha sido generado y expandido. Si es posible, cada arco tiene que etiquetarse con el nombre (y parámetros si hay) de la acción utilizada para expandir el nodo.
- Control de estados repetidos (convención):
  - En Amplitud no se generan nodos para estados que ya hayan sido generados antes.
  - En el resto, no se generan nodos con estados que estén en la rama que conduce al nodo que se está expandiendo.
    - Si por una rama ya he pasado por B, y puedo volver a generar B, se produce un ciclo el cual será peor respecto a la primera vez que llegué a B.
  - Para explicar cómo funciona, en estas soluciones a veces escribimos estos nodos, pero no es necesario en una respuesta de examen. En examen, aplicar sentido común y si borráis un nodo, explicad por qué.
- Para indicar cuál es la solución obtenida, hay que expresar la secuencia de acciones (si tienen nombre), así como el número de nodos generados y expandidos.

Para este grafo sin costes.

### Preguntas:

- Aplicar Búsqueda en Amplitud sobre el grafo.
- Aplicar Búsqueda en Profundidad sobre el grafo.

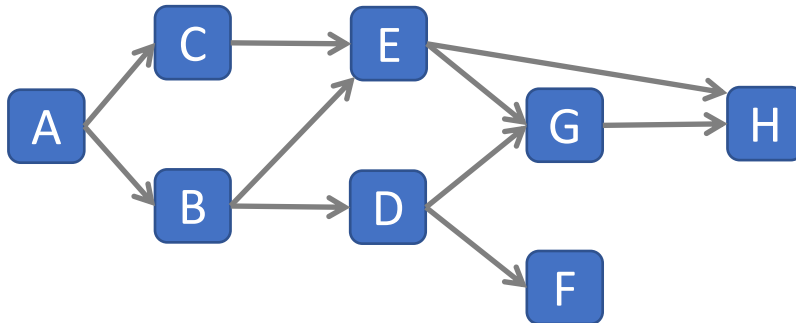
Para ambos indicar los nodos que se visitan en cada paso de los algoritmos. El orden de expansión de los nodos hijos es alfabético.



# Ejercicio 1

## Solución Amplitud

Grafo del ejercicio



Expansión de cada nodo



Resumen de expansión por ciclo

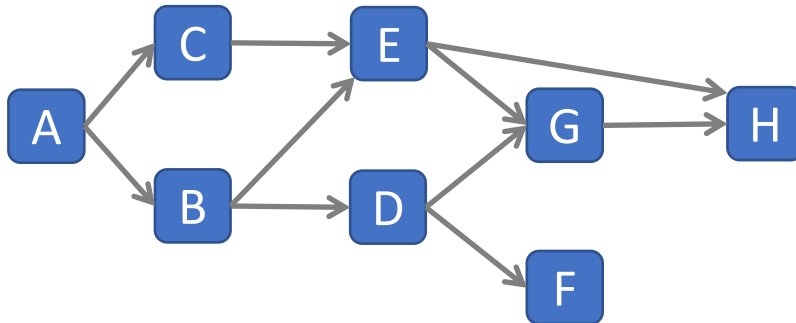
Ciclo	Expande	Genera	Abiertos
0	-	-	A



# Ejercicio 1

## Solución Amplitud

Grafo del ejercicio



Expansión de cada nodo



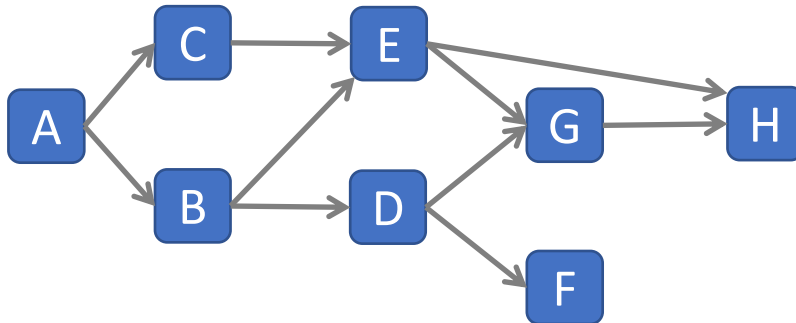
Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0	-	-	A
1	A	B, C	B, C

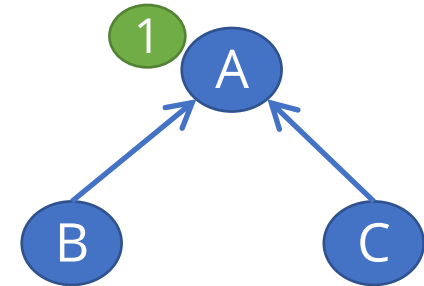
# Ejercicio 1

## Solución Amplitud

Grafo del ejercicio



Expansión de cada nodo



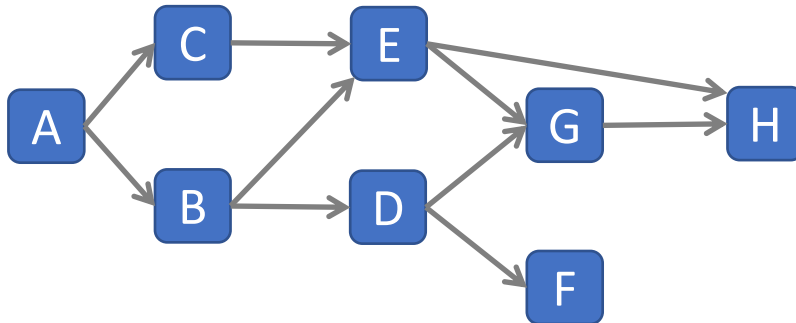
Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0	-	-	A
1	A	B, C	B, C

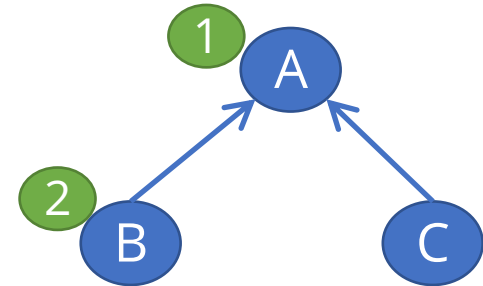
# Ejercicio 1

## Solución Amplitud

Grafo del ejercicio



Expansión de cada nodo



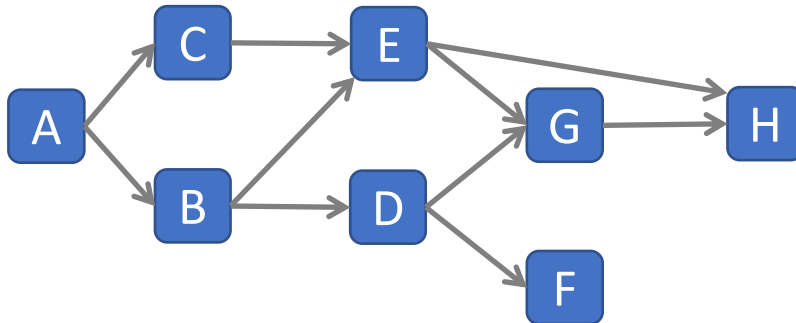
Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0	-	-	A
1	A	B, C	B, C
2	B	D, E	C, D, E

# Ejercicio 1

## Solución Amplitud

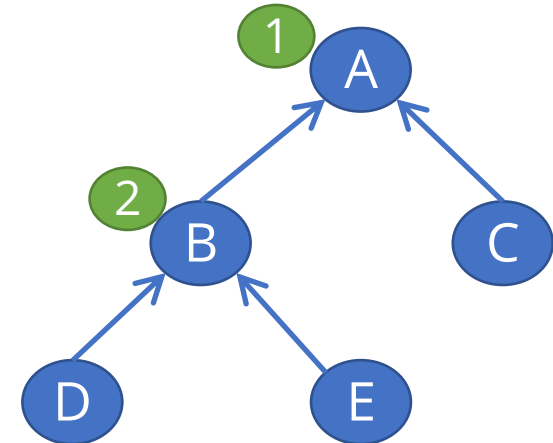
Grafo del ejercicio



Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0	-	-	A
1	A	B, C	B, C
2	B	D, E	C, D, E

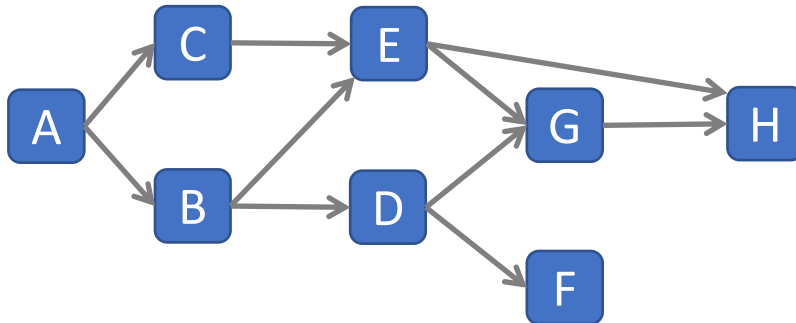
Expansión de cada nodo



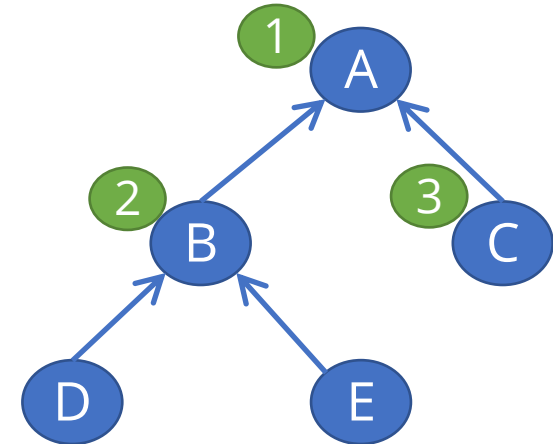
# Ejercicio 1

## Solución Amplitud

Grafo del ejercicio



Expansión de cada nodo



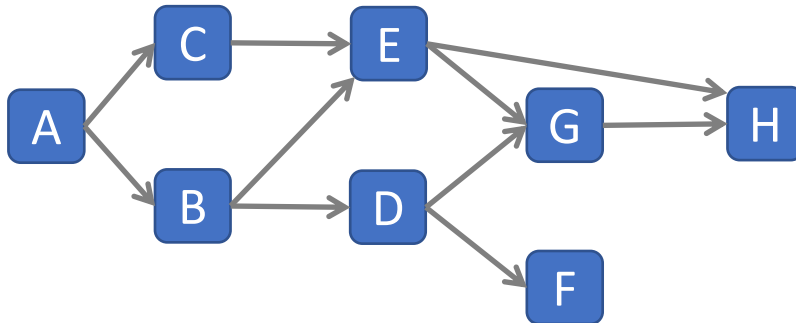
Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0	-	-	A
1	A	B, C	B, C
2	B	D, E	C, D, E
3	C	E	D, E

# Ejercicio 1

## Solución Amplitud

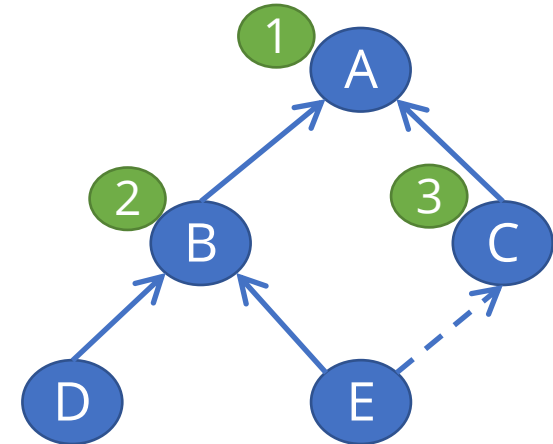
Grafo del ejercicio



Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0	-	-	A
1	A	B, C	B, C
2	B	D, E	C, D, E
3	C	E	D, E

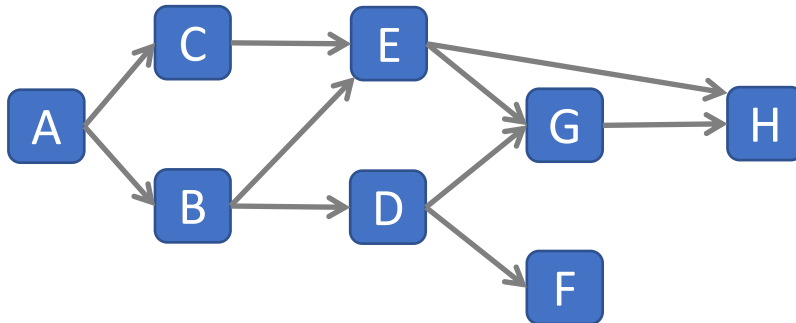
Expansión de cada nodo



# Ejercicio 1

## Solución Amplitud

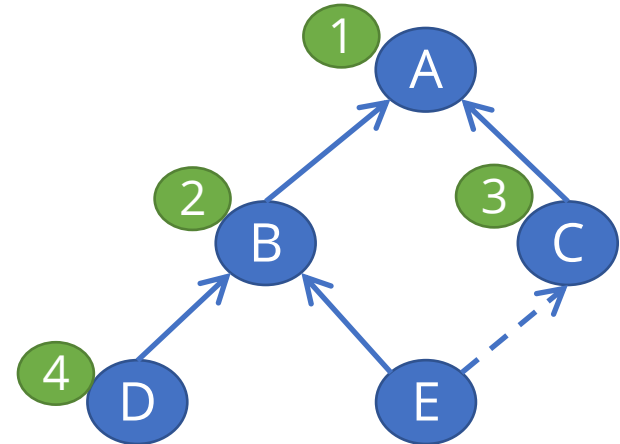
Grafo del ejercicio



Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0	-	-	A
1	A	B, C	B, C
2	B	D, E	C, D, E
3	C	E	D, E
4	D	F, G	E, F, G

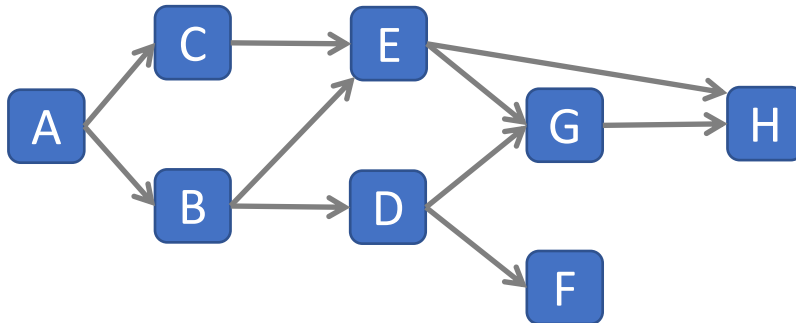
Expansión de cada nodo



# Ejercicio 1

## Solución Amplitud

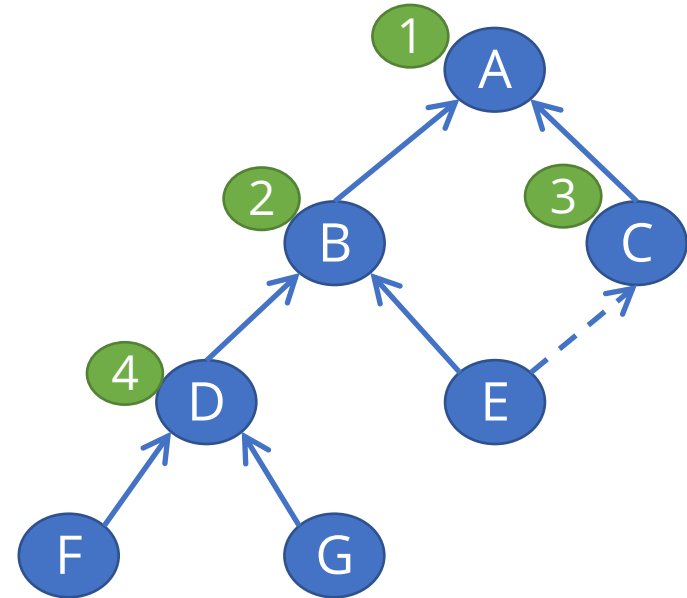
Grafo del ejercicio



Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0	-	-	A
1	A	B, C	B, C
2	B	D, E	C, D, E
3	C	E	D, E
4	D	F, G	E, F, G

Expansión de cada nodo

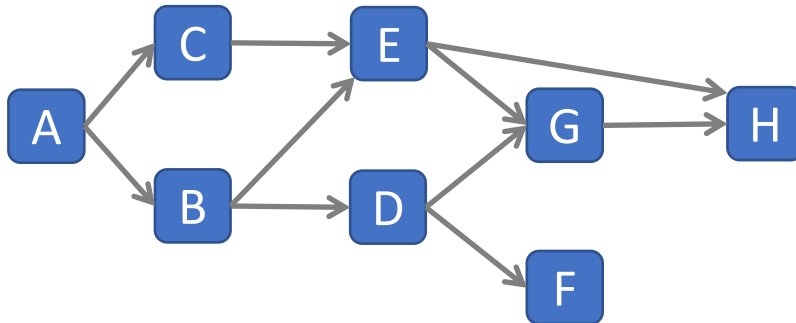




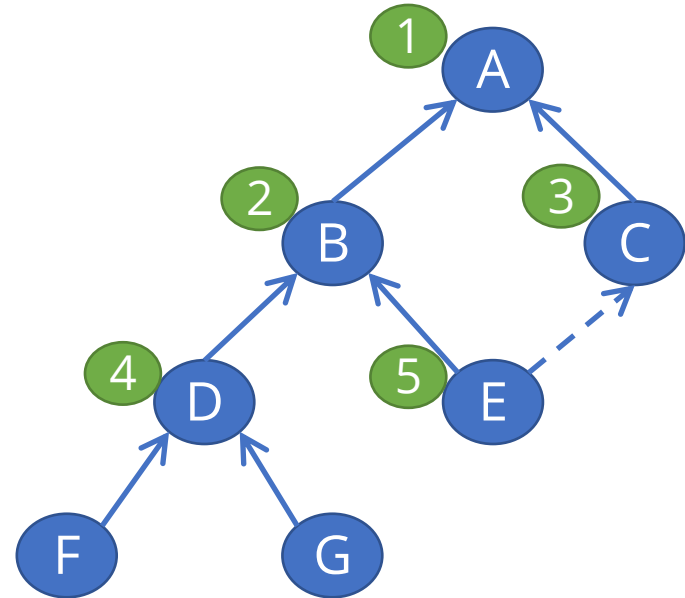
# Ejercicio 1

## Solución Amplitud

Grafo del ejercicio



Expansión de cada nodo



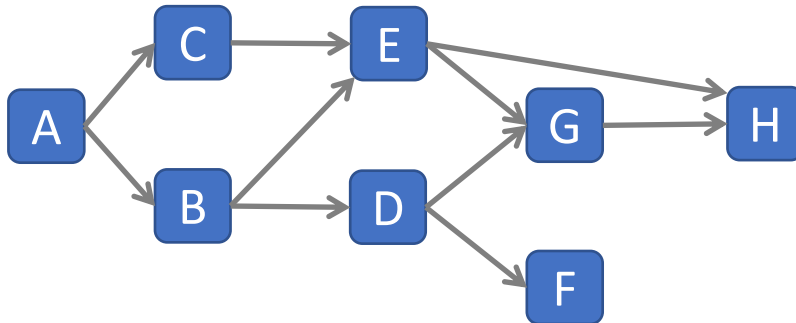
Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0	-	-	A
1	A	B, C	B, C
2	B	D, E	C, D, E
3	C	E	D, E
4	D	F, G	E, F, G
5	E	H	F, G, H

# Ejercicio 1

## Solución Amplitud

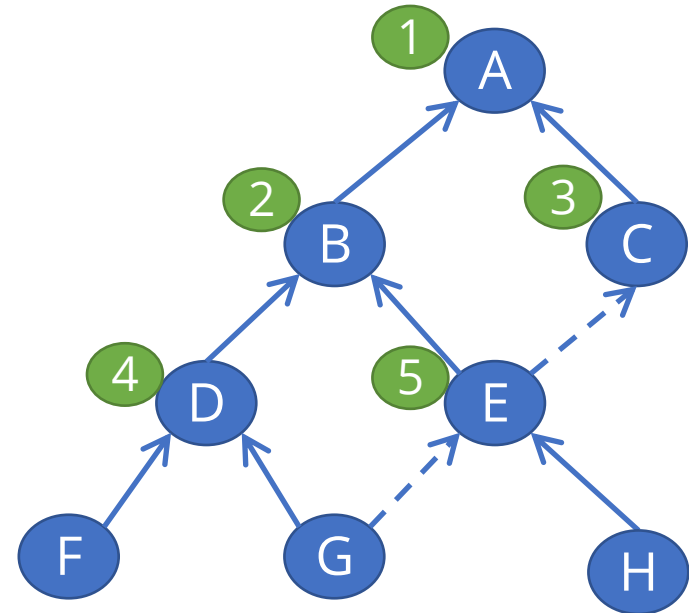
Grafo del ejercicio



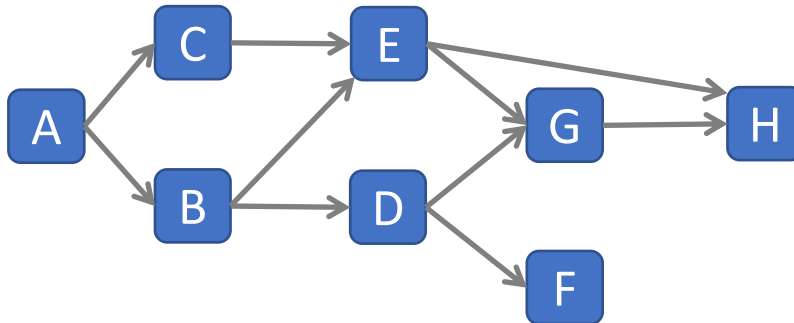
Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0	-	-	A
1	A	B, C	B, C
2	B	D, E	C, D, E
3	C	E	D, E
4	D	F, G	E, F, G
5	E	H	F, G, H

Expansión de cada nodo



Grafo del ejercicio



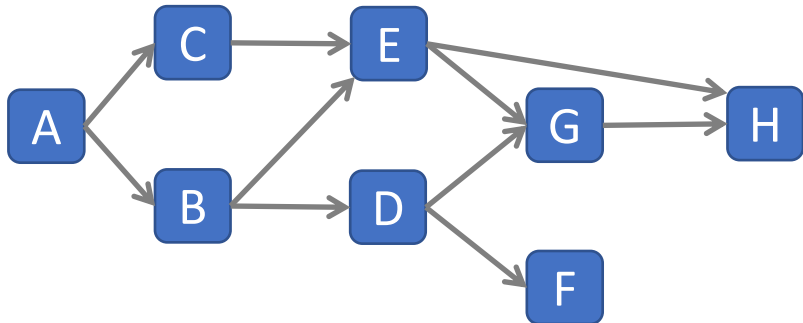
Expansión de cada nodo



Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0			A

Grafo del ejercicio



Expansión de cada nodo



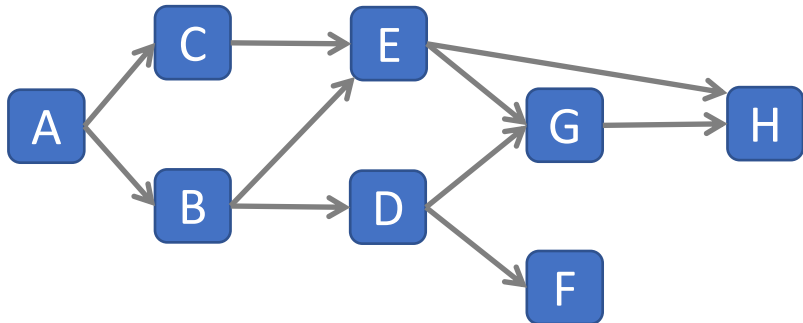
Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C	B, C

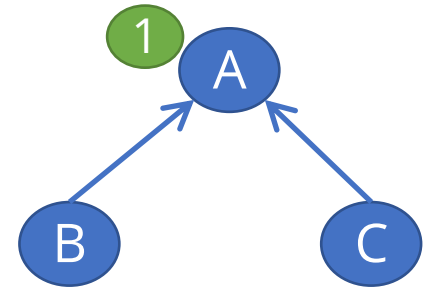
# Ejercicio 1

## Solución Profundidad

Grafo del ejercicio



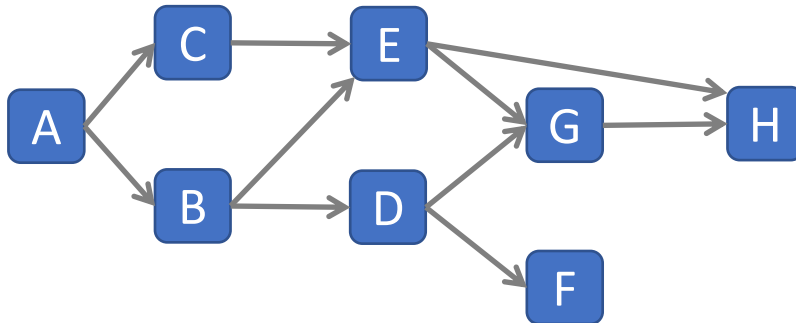
Expansión de cada nodo



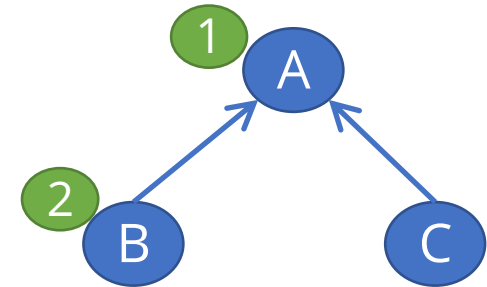
Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C	B, C

Grafo del ejercicio



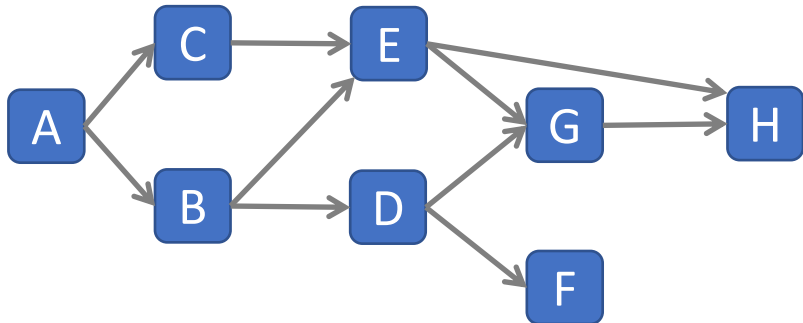
Expansión de cada nodo



Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C	B, C
2	B	D, E	D, E, C

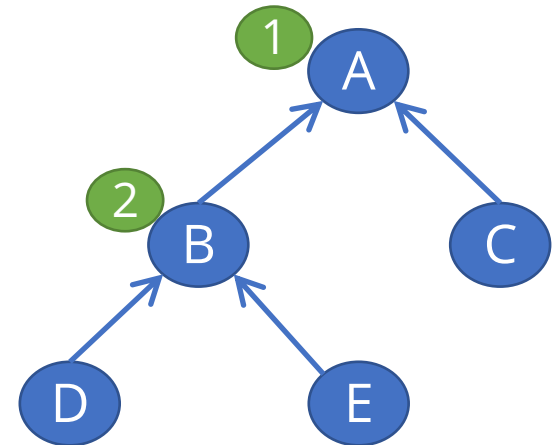
Grafo del ejercicio



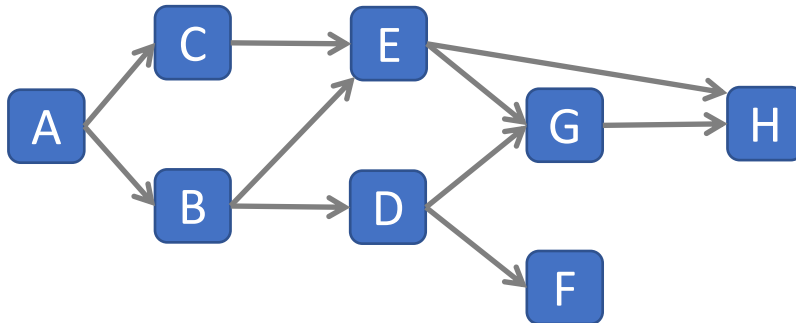
Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C	B, C
2	B	D, E	D, E, C

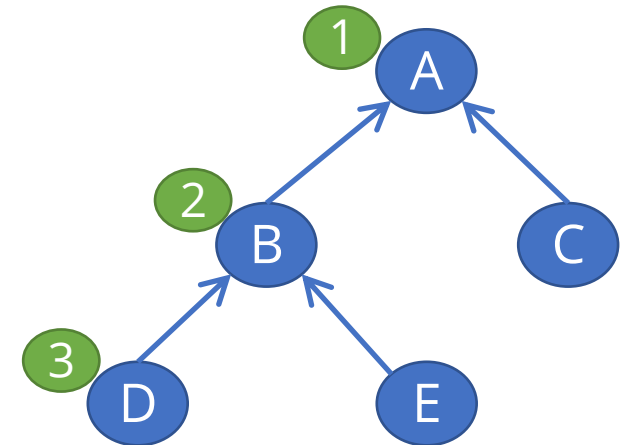
Expansión de cada nodo



Grafo del ejercicio



Expansión de cada nodo

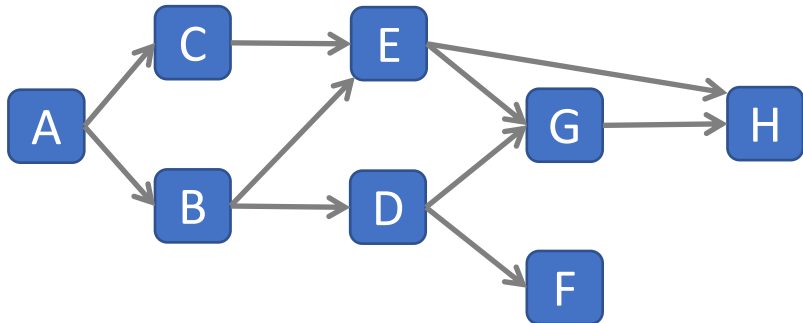


Resumen de expansión por ciclo

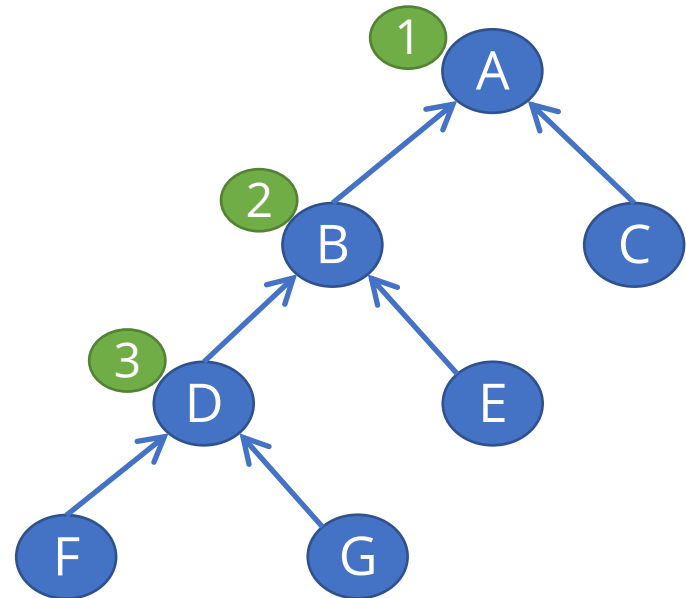
Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C	B, C
2	B	D, E	D, E, C
3	D	F, G	F, G, E, C



Grafo del ejercicio



Expansión de cada nodo



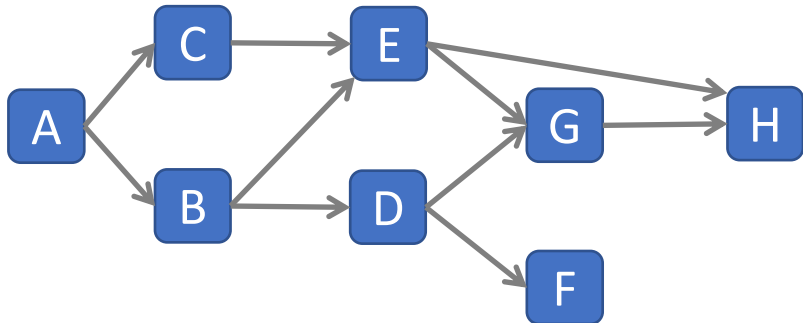
Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C	B, C
2	B	D, E	D, E, C
3	D	F, G	F, G, E, C

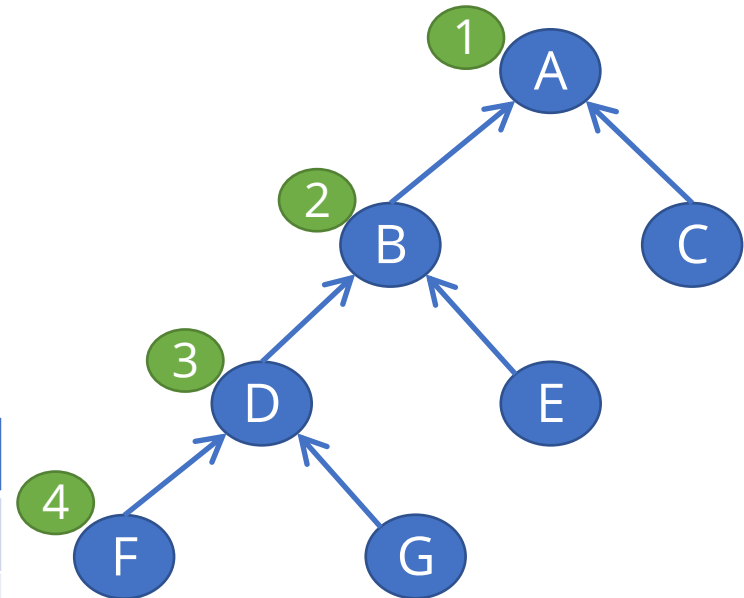
# Ejercicio 1

## Solución Profundidad

Grafo del ejercicio



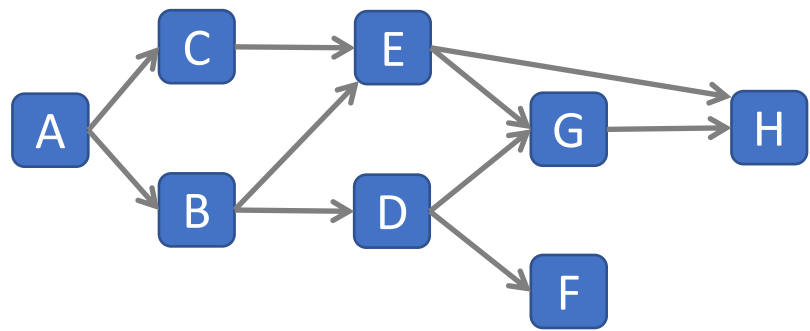
Expansión de cada nodo



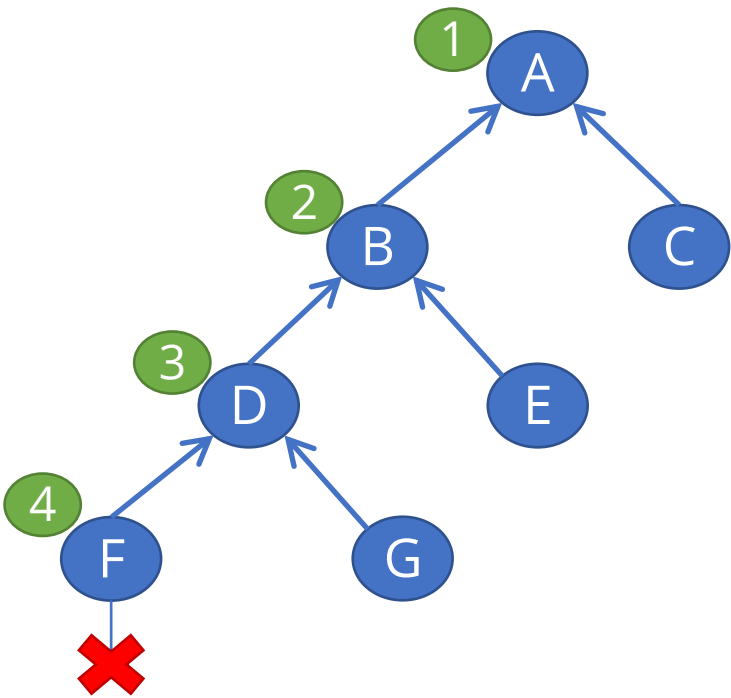
Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C	B, C
2	B	D, E	D, E, C
3	D	F, G	F, G, E, C
4	F	-	G, E, C

Grafo del ejercicio



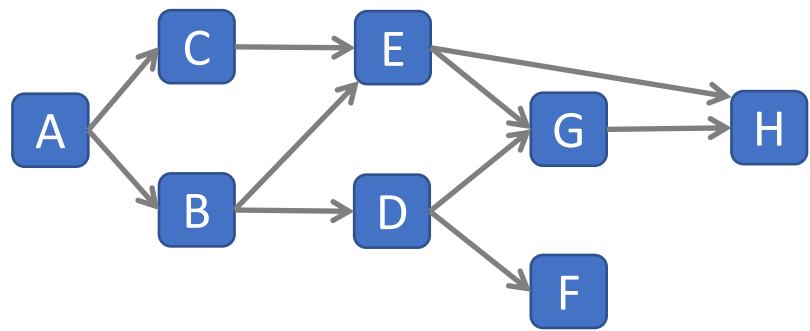
Expansión de cada nodo



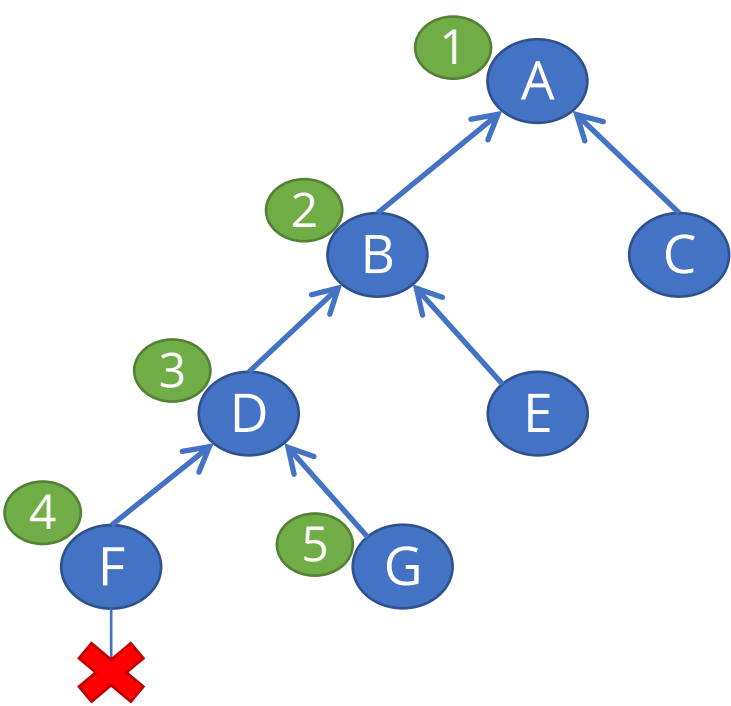
Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C	B, C
2	B	D, E	D, E, C
3	D	F, G	F, G, E, C
4	F	-	G, E, C

Grafo del ejercicio



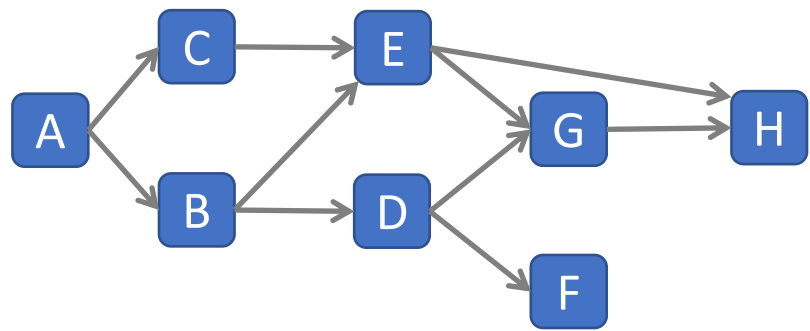
Expansión de cada nodo



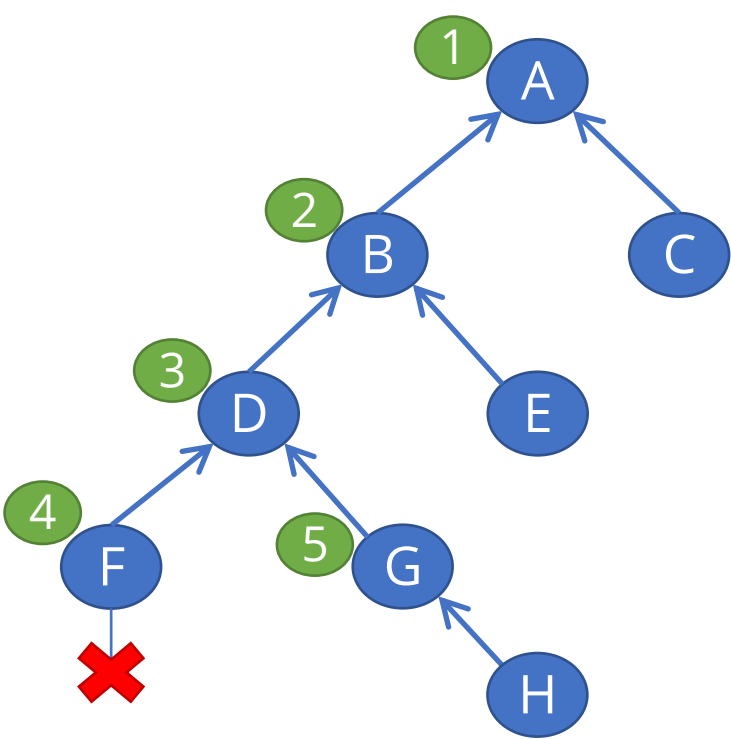
Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C	B, C
2	B	D, E	D, E, C
3	D	F, G	F, G, E, C
4	F	-	G, E, C
5	G	H	H, E, C

Grafo del ejercicio



Expansión de cada nodo



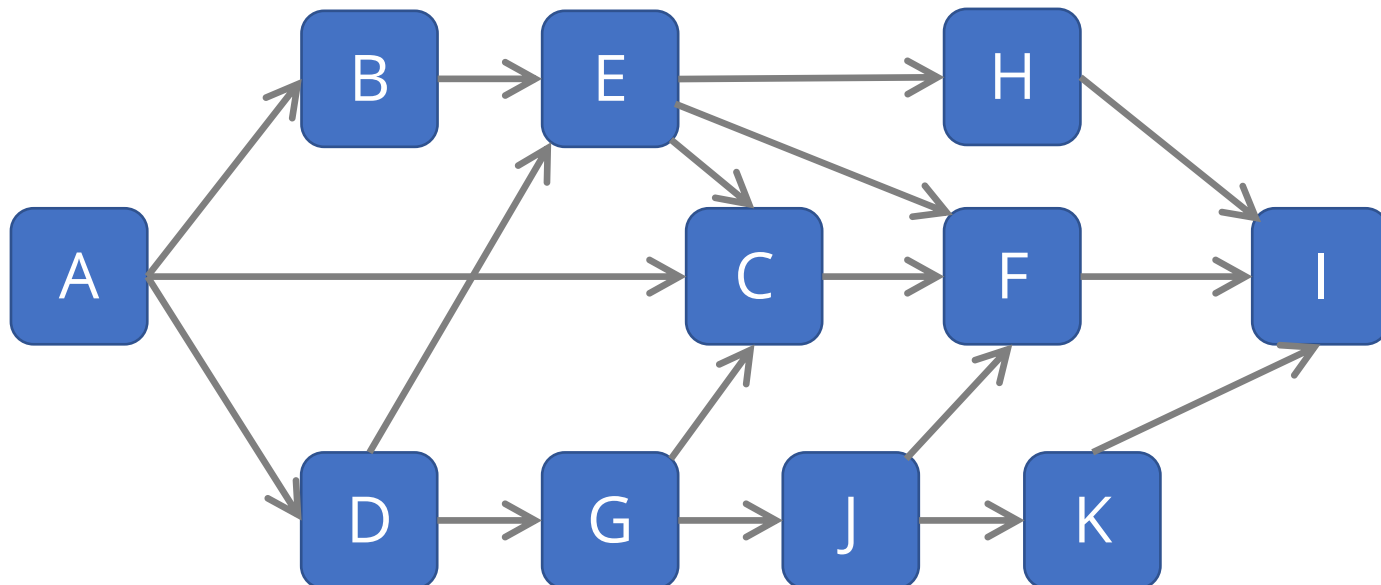
Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C	B, C
2	B	D, E	D, E, C
3	D	F, G	F, G, E, C
4	F	-	G, E, C
5	G	H	H, E, C

Para este grafo.

**Preguntas:**

- ¿Qué algoritmo de búsqueda encontrará el camino mínimo entre A e I?
- Encontrar dicho camino. Indicar los nodos que se visitan en cada paso. El orden de expansión de los nodos hijos es alfabético.



## ¿Qué algoritmo de búsqueda encontrará el camino mínimo entre A e I?

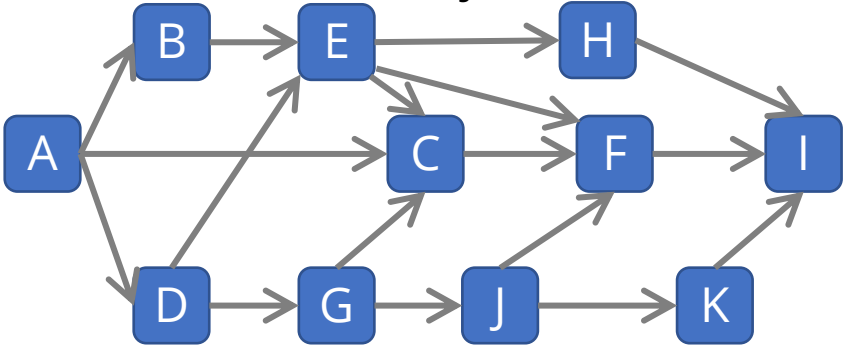
Si se analizan las estrategias vistas en clase, se observa que:

- Por las características de la búsqueda en profundidad no garantiza llegar al nodo I por el camino más corto. Encontrará una solución, pero no tiene por qué ser la mejor (por ende el camino más corto).
- Búsqueda en amplitud sí lo encontrará, al explorar todos los caminos más cortos antes de pasar a los de una longitud mayor.

**¿Qué algoritmo de búsqueda encontrará el camino mínimo entre A e I?**



Grafo del ejercicio



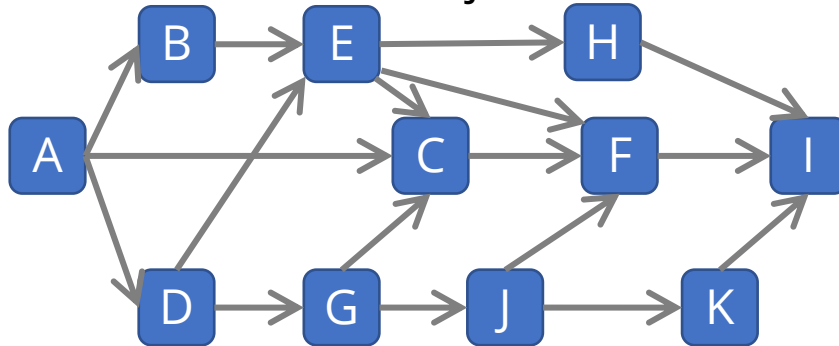
Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0			A

Expansión de cada nodo



Grafo del ejercicio

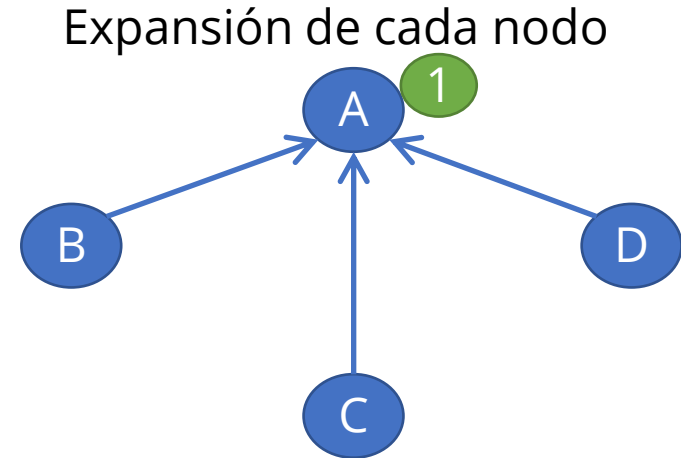
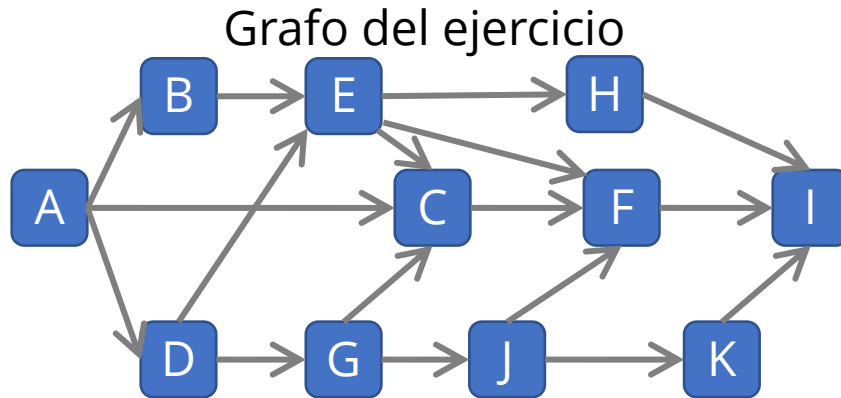


Expansión de cada nodo



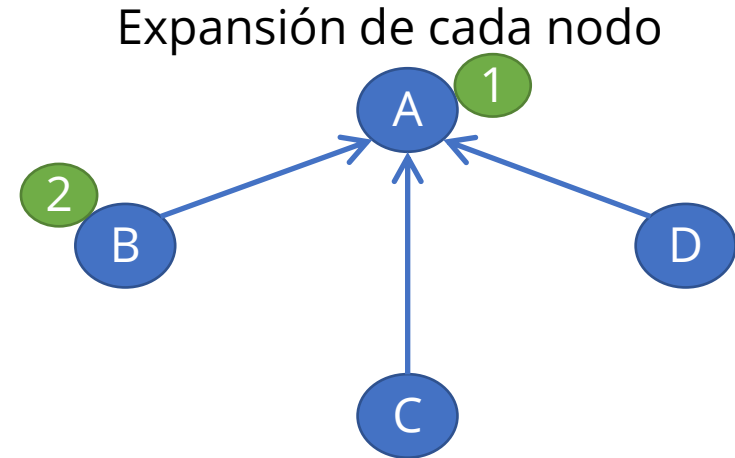
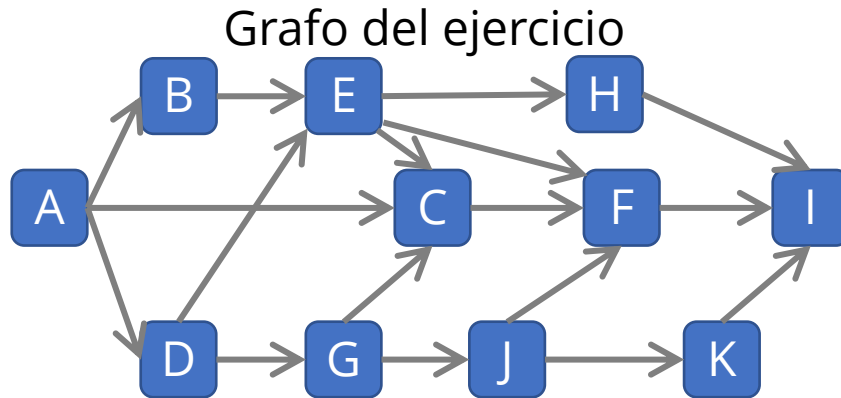
Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C, D	B, C, D



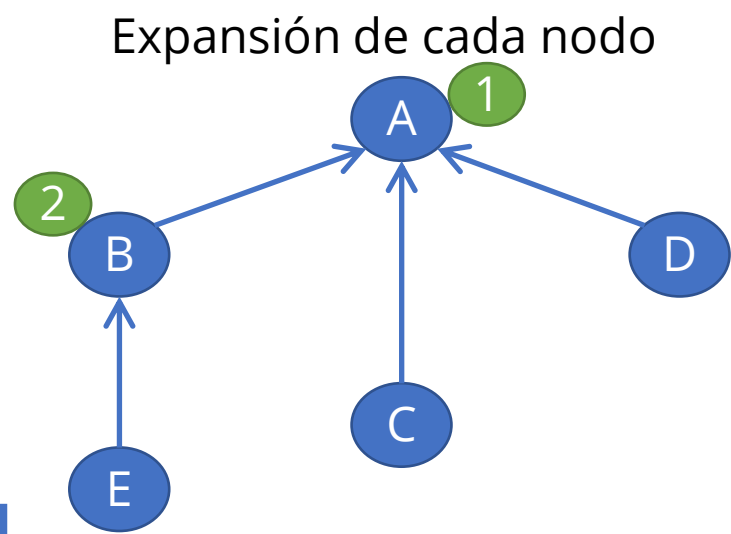
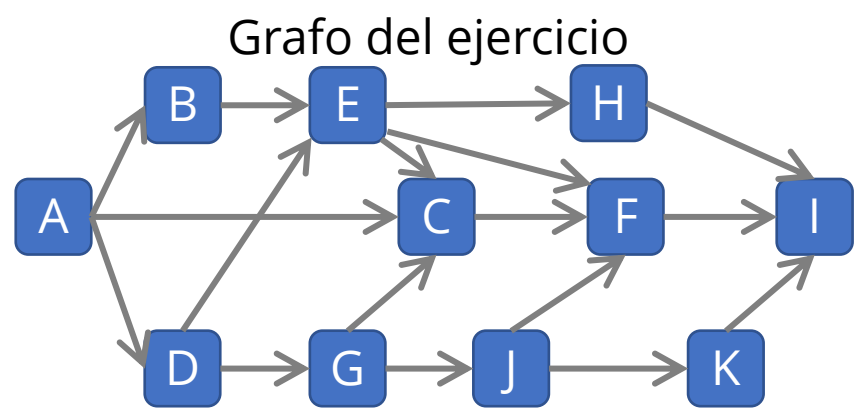
Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C, D	B, C, D



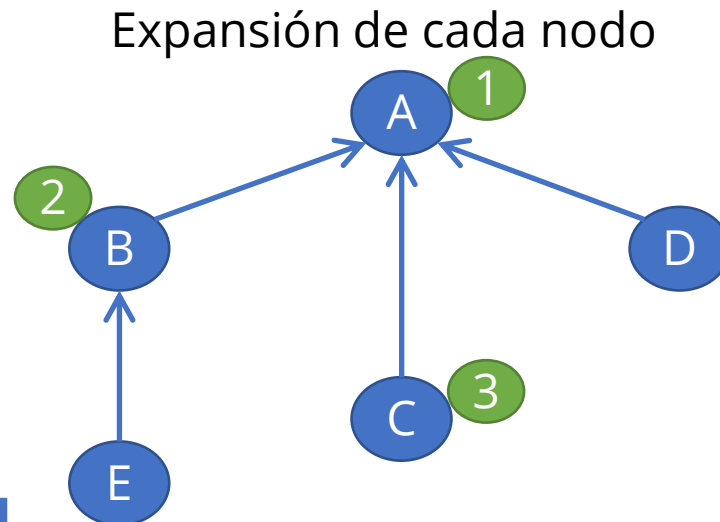
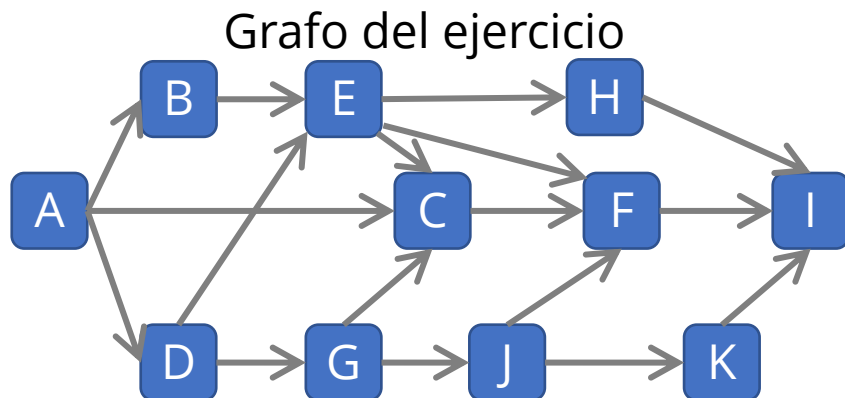
Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C, D	B, C, D
2	B	E	C, D, E



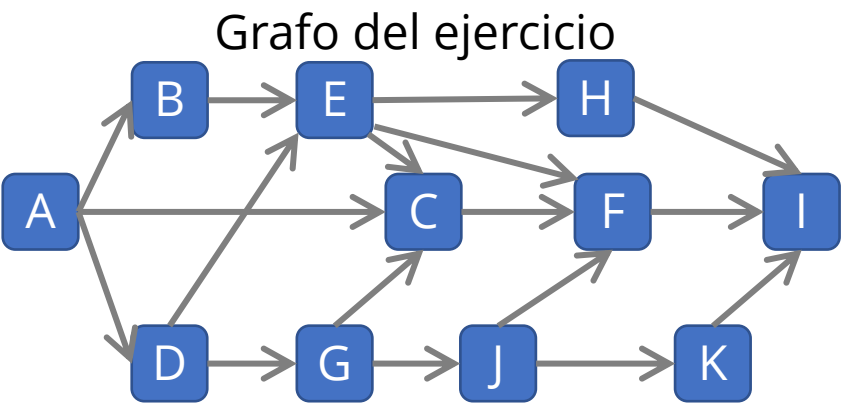
Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C, D	B, C, D
2	B	E	C, D, E



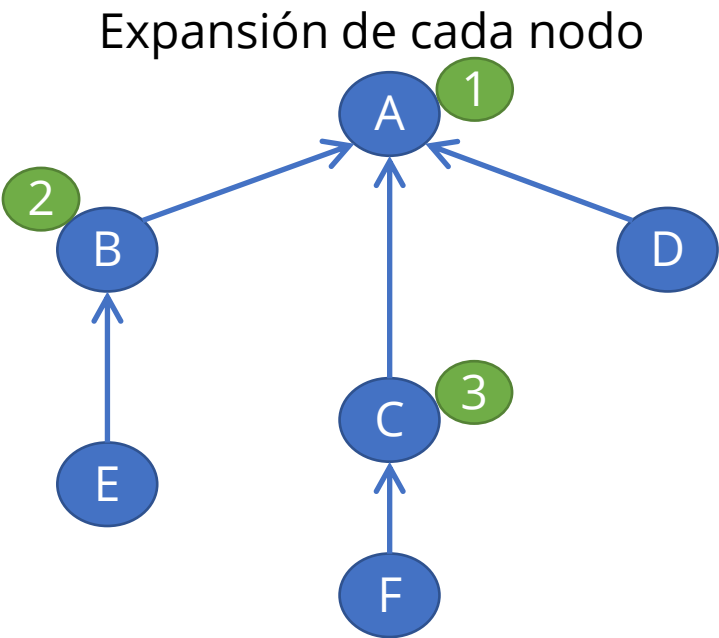
Resumen de expansión por ciclo

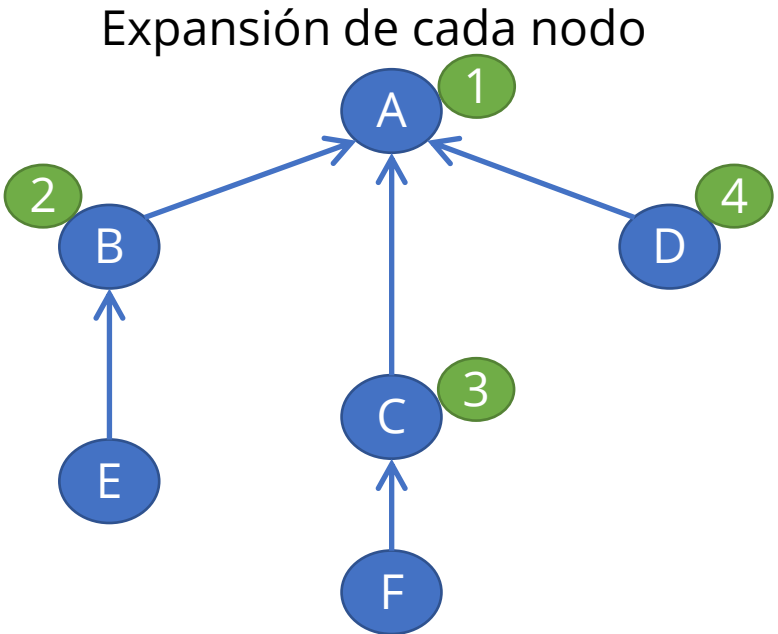
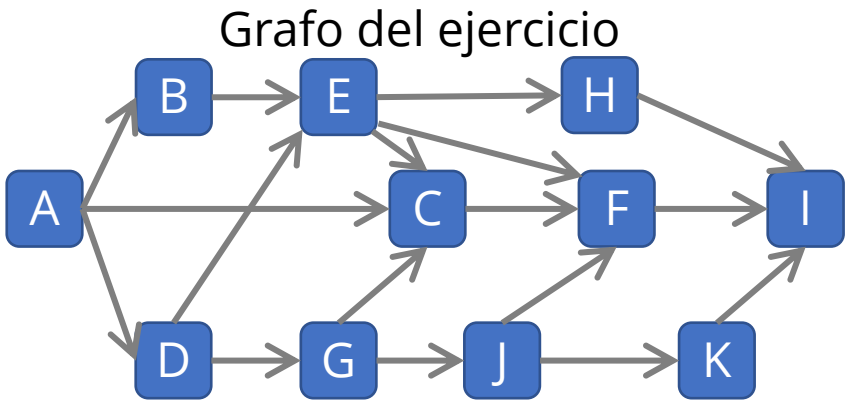
Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C, D	B, C, D
2	B	E	C, D, E
3	C	F	D, E, F



Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C, D	B, C, D
2	B	E	C, D, E
3	C	F	D, E, F

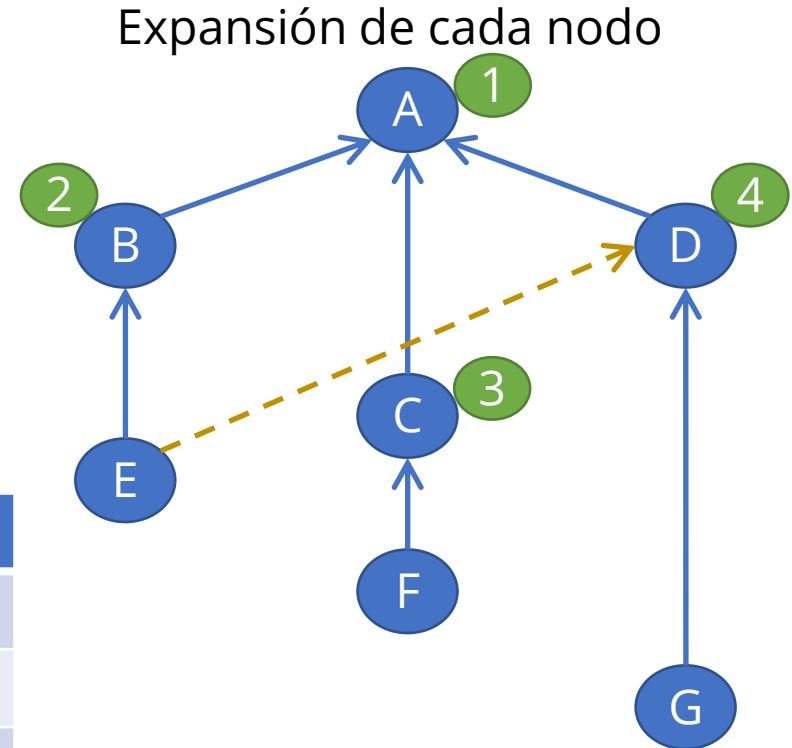
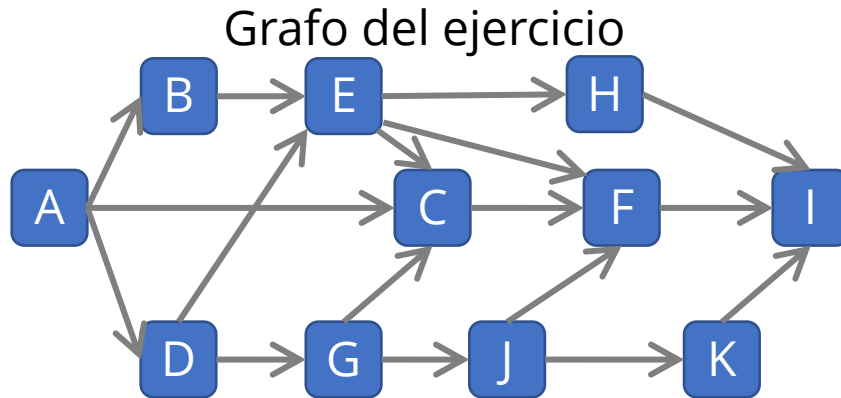




Resumen de expansión por ciclo

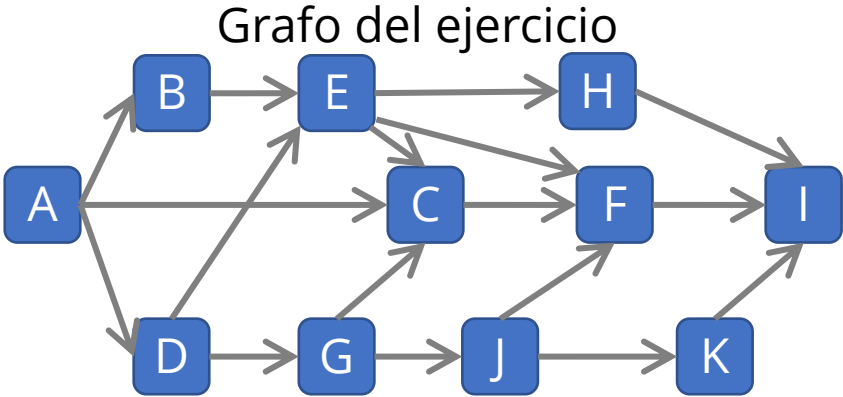
Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C, D	B, C, D
2	B	E	C, D, E
3	C	F	D, E, F
4	D	G	E, F, G





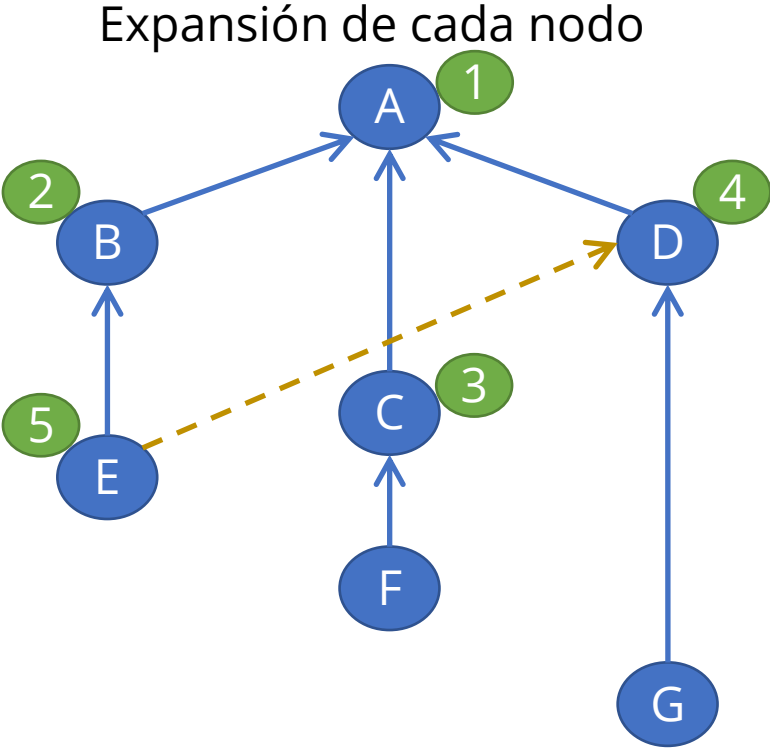
Resumen de expansión por ciclo

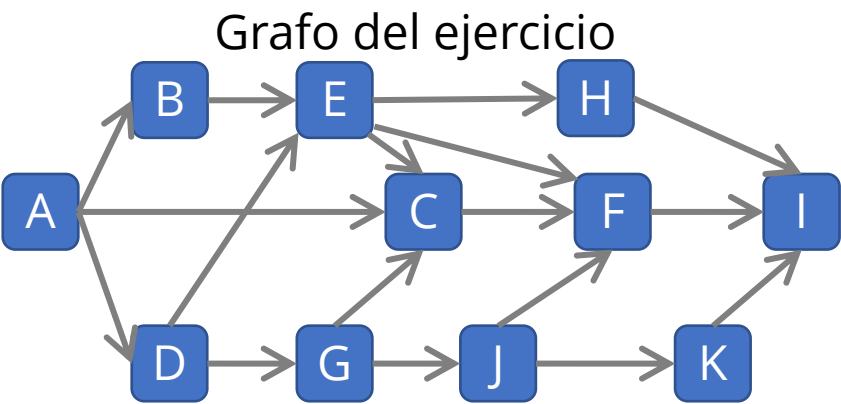
Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C, D	B, C, D
2	B	E	C, D, E
3	C	F	D, E, F
4	D	G	E, F, G



Resumen de expansión por ciclo

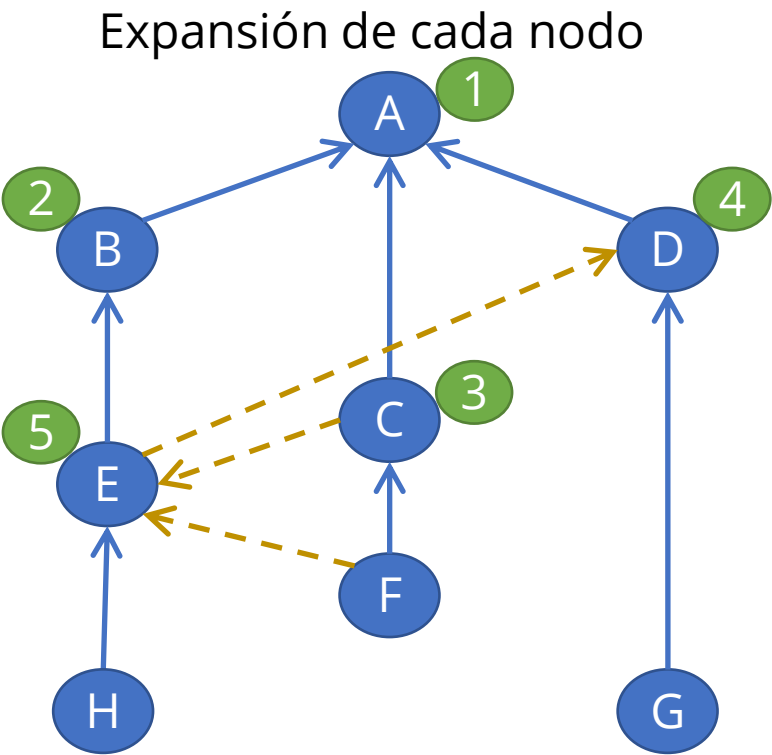
Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C, D	B, C, D
2	B	E	C, D, E
3	C	F	D, E, F
4	D	G	E, F, G
5	E	H	F, G, H

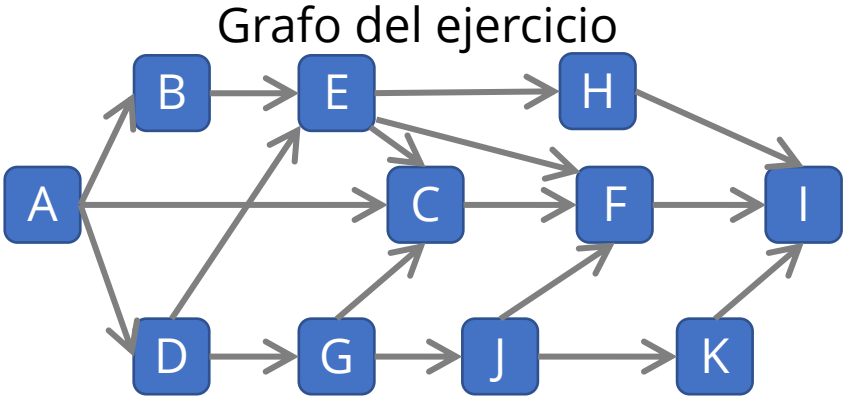




Resumen de expansión por ciclo

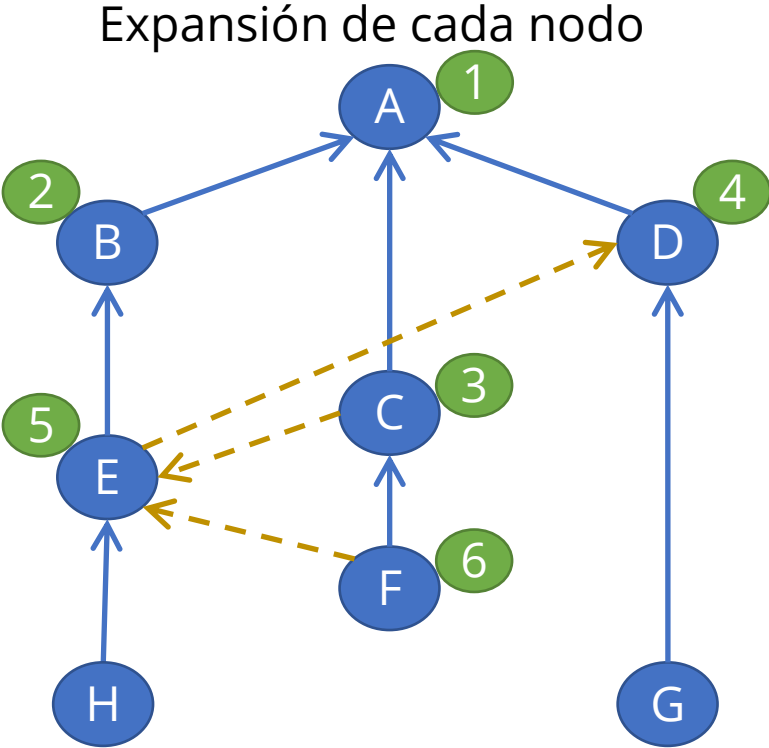
Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C, D	B, C, D
2	B	E	C, D, E
3	C	F	D, E, F
4	D	G	E, F, G
5	E	H	F, G, H

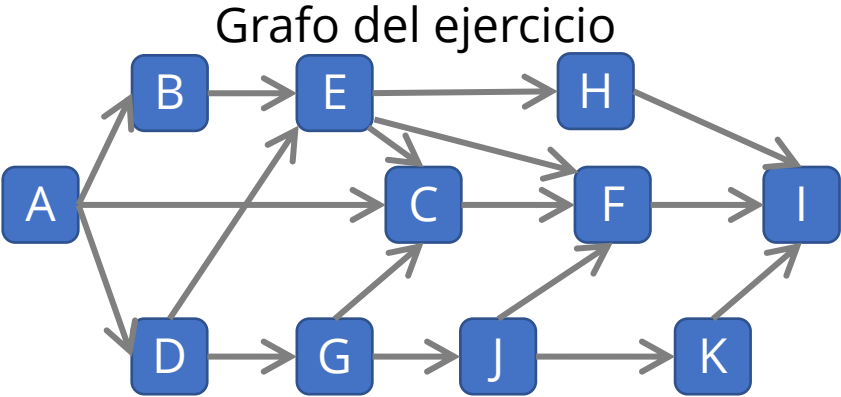




Resumen de expansión por ciclo

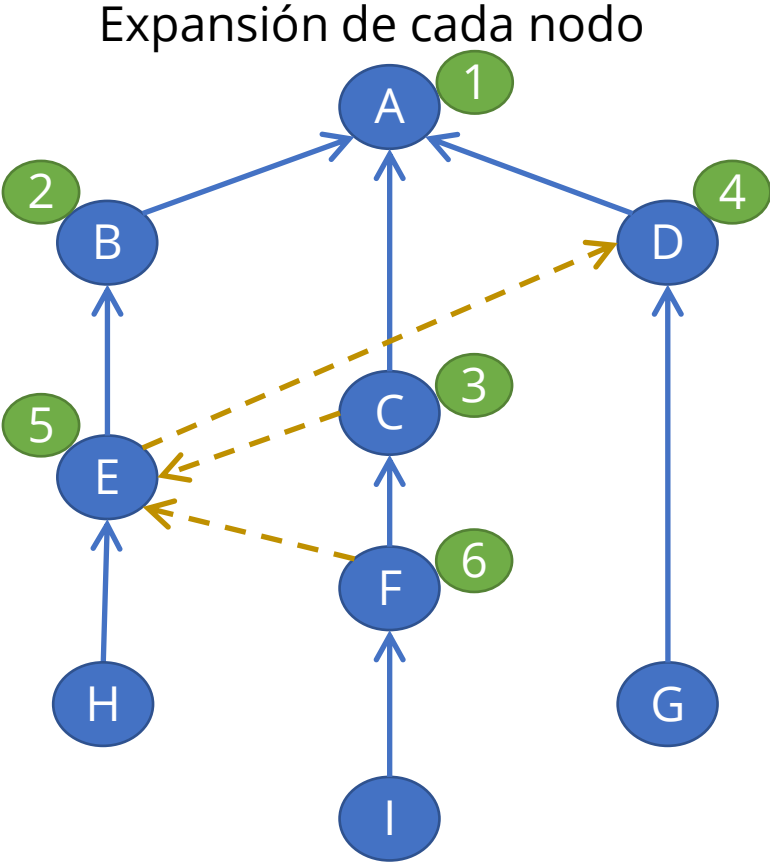
Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C, D	B, C, D
2	B	E	C, D, E
3	C	F	D, E, F
4	D	G	E, F, G
5	E	H	F, G, H
6	F	I	G, H, I





Resumen de expansión por ciclo

Ciclo	Expande	Genera	Abiertos
0			A
1	A	B, C, D	B, C, D
2	B	E	C, D, E
3	C	F	D, E, F
4	D	G	E, F, G
5	E	H	F, G, H
6	F	I	G, H, I



Un pastor P tiene que pasar tres cosas de una orilla a la otra de un río.

- un lobo L
- una oveja O
- una col C

Dispone de una barca en la que sólo caben él y una de las otras tres cosas a la vez.

Si el lobo se queda sin el pastor con la oveja se la come, si la oveja se queda sin el pastor con la col se la come.

¿Cómo debe hacerlo realizar los movimientos sin que se coman nada?

**Preguntas:**

- ¿Cómo se puede representar cada estado?
- ¿Cuáles son los operadores para la representación escogida?
- ¿Cuáles son los estados generados a partir del estado inicial?
- ¿Cómo se desarrollaría tras los primeros 5 movimientos?

**¿Cómo se puede representar cada estado?**

**¿Cuáles son los operadores para la representación escogida?**

---

**¿Cómo se puede representar cada estado?**

- Cada estado del problema se puede representar según las cosas que hay a cada lado del río. (<objetosIzq>; <objetosDer>)
- Los objetos podrán ser tanto el pastor P(pastor) como los elementos que transporta: L(lobo), O(oveja), C(col). Si un lado está vacío, se representa con N(nada).
- El estado inicial por tanto será: (P,L,O,C; N)
- El estado final por tanto será: (N; P,L,O,C)

**¿Cuáles son los operadores para la representación escogida?**



### ¿Cómo se puede representar cada estado?

- Cada estado del problema se puede representar según las cosas que hay a cada lado del río. (<objetosIzq>; <objetosDer>)
- Los objetos podrán ser tanto el pastor P(pastor) como los elementos que transporta: L(lobo), O(oveja), C(col). Si un lado está vacío, se representa con N(nada).
- El estado inicial por tanto será: (P,L,O,C; N)
- El estado final por tanto será: (N; P,L,O,C)

### ¿Cuáles son los operadores para la representación escogida?

Las acciones serán el movimiento de objetos de un lado a otro:

- D(X): Mueve un elemento de izquierda a derecha.
  - Es necesario que X y P estén a la izquierda.
  - No puede quedar en el lado izquierdo sin el pastor: L+O, O+C
- I(X): Mueve un elemento de derecha a izquierda.
  - Es necesario que X y P estén a la derecha.
  - No puede quedar en el lado derecho sin el pastor: L+O, O+C

---

**¿Cuáles son los estados generados a partir del estado inicial?**

**¿Cómo se desarrollaría tras los primeros 5 movimientos?**

Leyenda

Combinación  
restringida

Combinación  
repetida

Combinación  
válida

### ¿Cuáles son los estados generados a partir del estado inicial?

Se pueden realizar movimientos a derechas:

- D(nada) resultaría en (LOC, P) dejaría en el lado derecho L+O y O+C
- D(L) resultaría en (OC, PL) dejaría en el lado derecho O+C
- D(O) resultaría en (LC, PO) y no fallaría ninguna restricción
- D(C) resultaría en (LO, PC) dejaría en el lado derecho L+O

### ¿Cómo se desarrollaría tras los primeros 5 movimientos?

(PLOC; N)

Leyenda

Combinación  
restringida

Combinación  
repetida

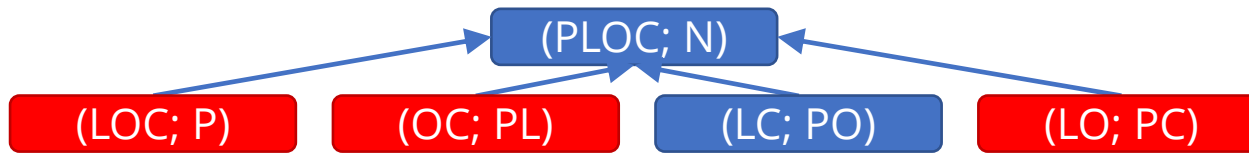
Combinación  
válida

## ¿Cuáles son los estados generados a partir del estado inicial?

Se pueden realizar movimientos a derechas:

- D(nada) resultaría en (LOC, P) dejaría en el lado derecho L+O y O+C
- D(L) resultaría en (OC, PL) dejaría en el lado derecho O+C
- D(O) resultaría en (LC, PO) y no fallaría ninguna restricción
- D(C) resultaría en (LO, PC) dejaría en el lado derecho L+O

## ¿Cómo se desarrollaría tras los primeros 5 movimientos?



### Leyenda

Combinación restringida

Combinación repetida

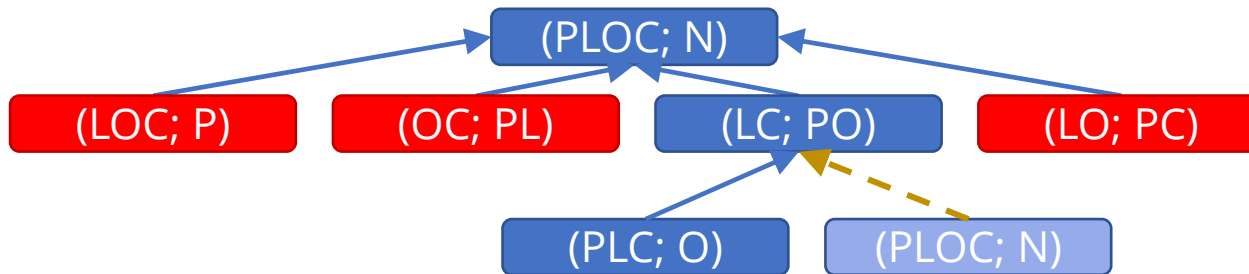
Combinación válida

## ¿Cuáles son los estados generados a partir del estado inicial?

Se pueden realizar movimientos a derechas:

- D(nada) resultaría en (LOC, P) dejaría en el lado derecho L+O y O+C
- D(L) resultaría en (OC, PL) dejaría en el lado derecho O+C
- D(O) resultaría en (LC, PO) y no fallaría ninguna restricción
- D(C) resultaría en (LO, PC) dejaría en el lado derecho L+O

## ¿Cómo se desarrollaría tras los primeros 5 movimientos?



### Leyenda

Combinación restringida

Combinación repetida

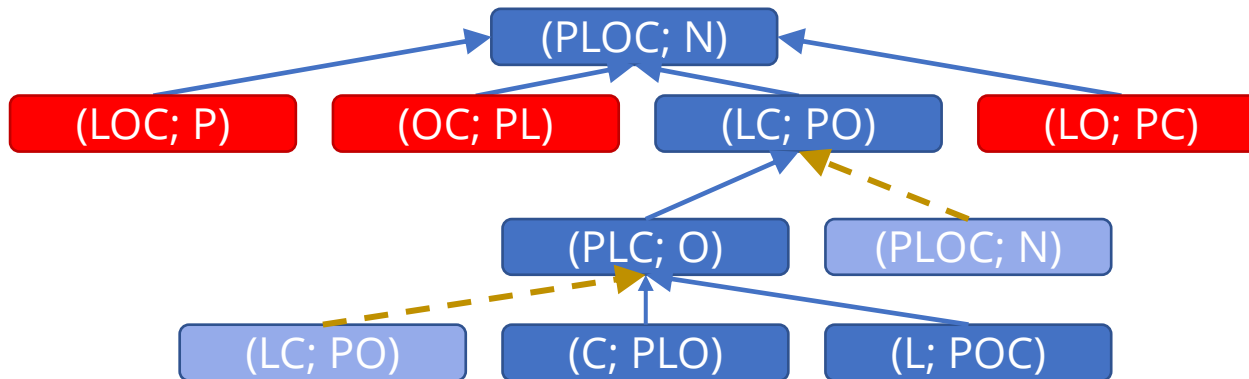
Combinación válida

## ¿Cuáles son los estados generados a partir del estado inicial?

Se pueden realizar movimientos a derechas:

- D(nada) resultaría en (LOC, P) dejaría en el lado derecho L+O y O+C
- D(L) resultaría en (OC, PL) dejaría en el lado derecho O+C
- D(O) resultaría en (LC, PO) y no fallaría ninguna restricción
- D(C) resultaría en (LO, PC) dejaría en el lado derecho L+O

## ¿Cómo se desarrollaría tras los primeros 5 movimientos?



### Leyenda

Combinación restringida

Combinación repetida

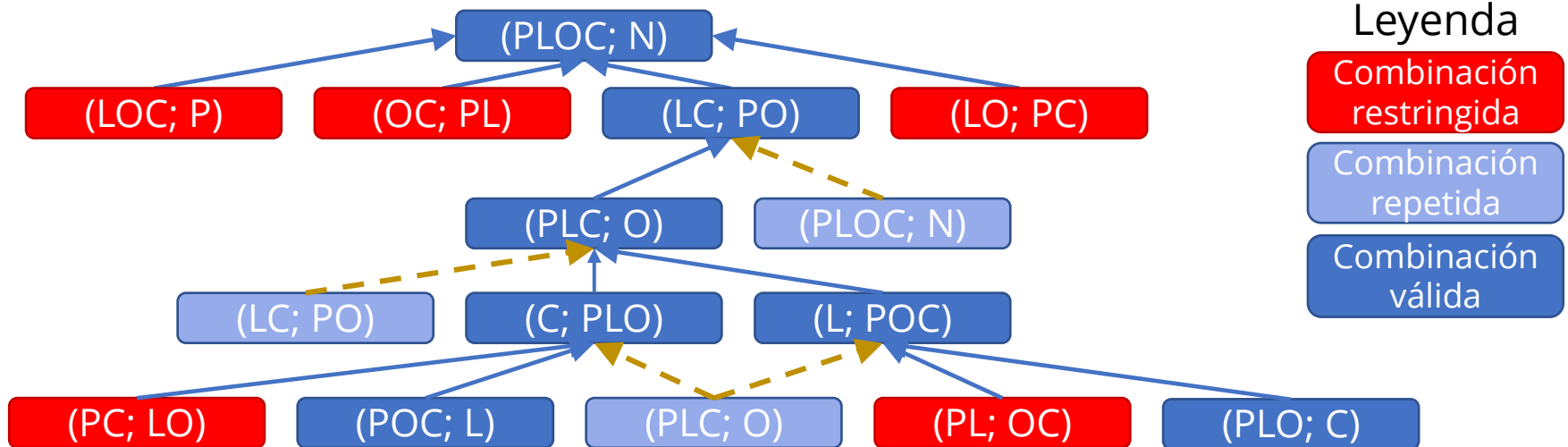
Combinación válida

## ¿Cuáles son los estados generados a partir del estado inicial?

Se pueden realizar movimientos a derechas:

- D(nada) resultaría en (LOC, P) dejaría en el lado derecho L+O y O+C
- D(L) resultaría en (OC, PL) dejaría en el lado derecho O+C
- D(O) resultaría en (LC, PO) y no fallaría ninguna restricción
- D(C) resultaría en (LO, PC) dejaría en el lado derecho L+O

## ¿Cómo se desarrollaría tras los primeros 5 movimientos?

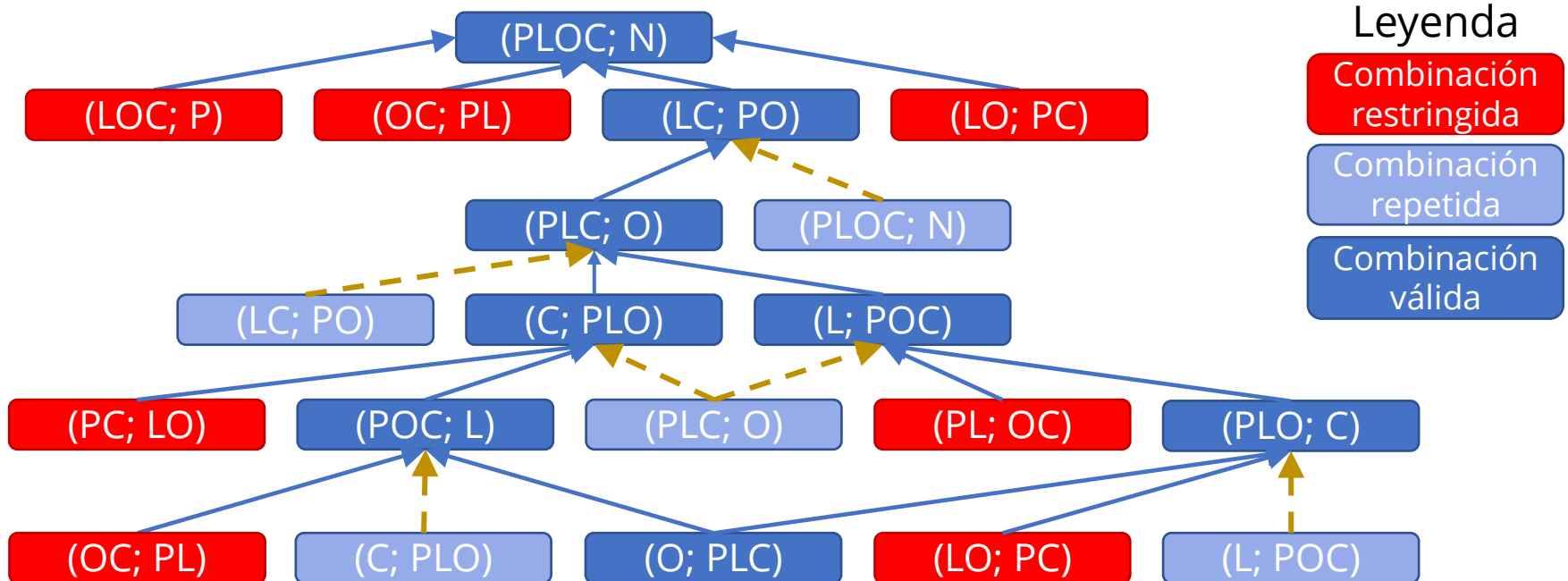


## ¿Cuáles son los estados generados a partir del estado inicial?

Se pueden realizar movimientos a derechas:

- D(nada) resultaría en (LOC, P) dejaría en el lado derecho L+O y O+C
- D(L) resultaría en (OC, PL) dejaría en el lado derecho O+C
- D(O) resultaría en (LC, PO) y no fallaría ninguna restricción
- D(C) resultaría en (LO, PC) dejaría en el lado derecho L+O

## ¿Cómo se desarrollaría tras los primeros 5 movimientos?

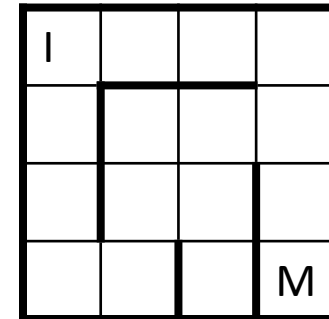




El objetivo consiste en encontrar el camino desde el origen I hasta la meta M

Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



### Preguntas:

- Tomando en cuenta el orden de los operadores, hacer búsqueda en amplitud, marcando en cada casilla el número que indica en qué paso el nodo se expande (es decir, cuando se generan sus sucesores).
- Hacer búsqueda en profundidad de igual forma.
- Cambiar el orden de los operadores para que la búsqueda en profundidad encuentre la meta más rápido.

Operadores (en orden de prioridad):

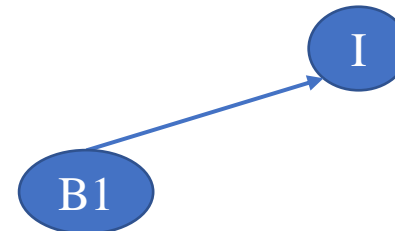
1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



	1	2	3	4
A	I			
B				
C				
D				M

Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha

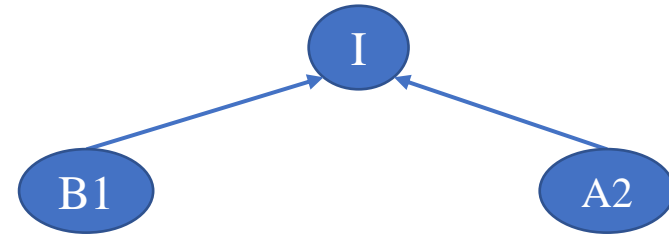


	1	2	3	4
A	I			
B	↓ 1			
C				
D				M

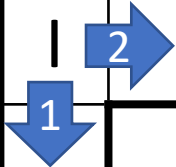
# Solución en amplitud

Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha

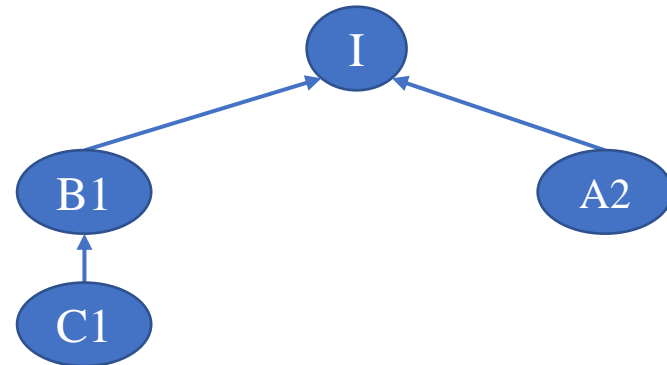
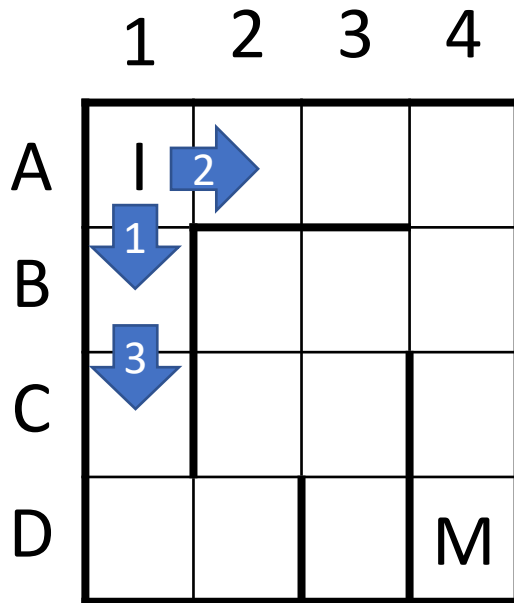


	1	2	3	4
A	I			
B				
C				
D				M



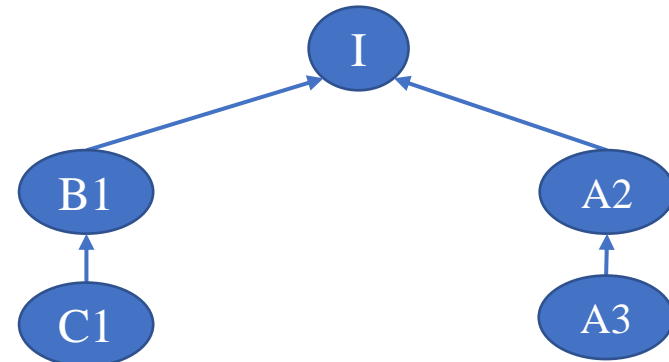
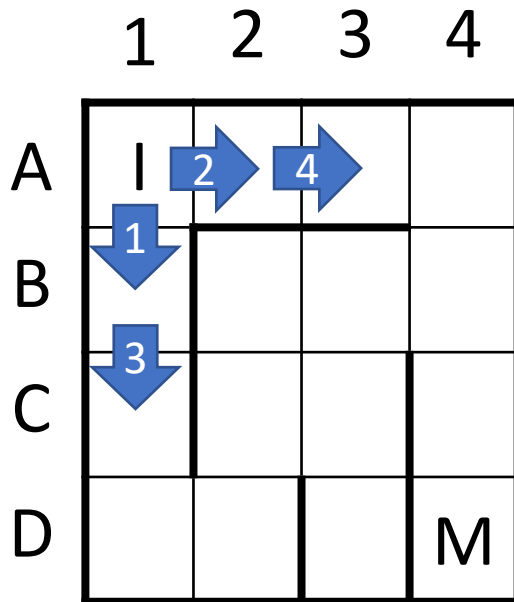
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



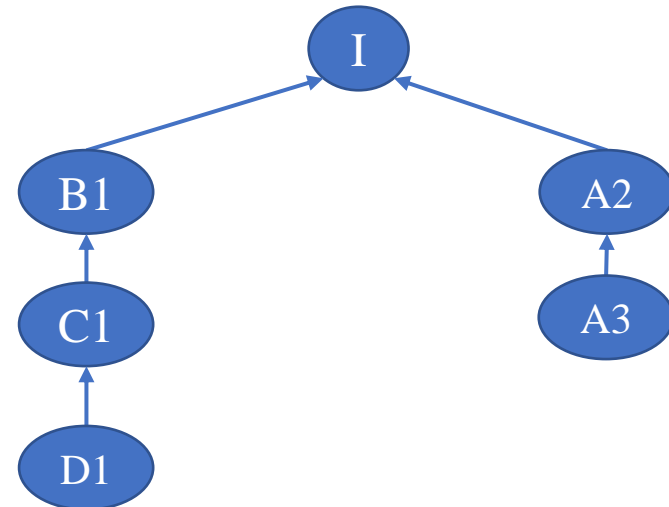
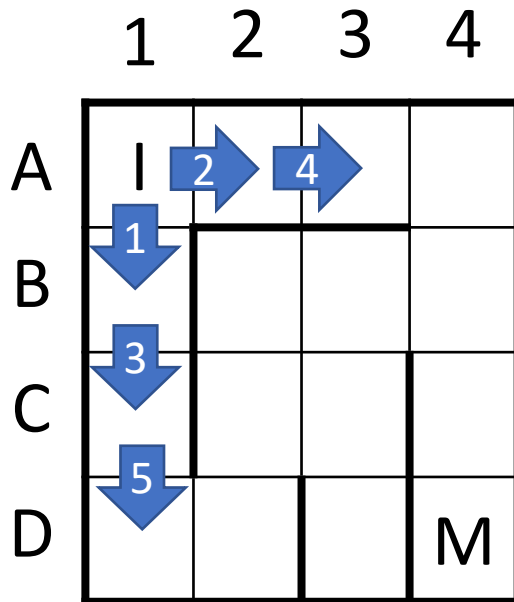
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



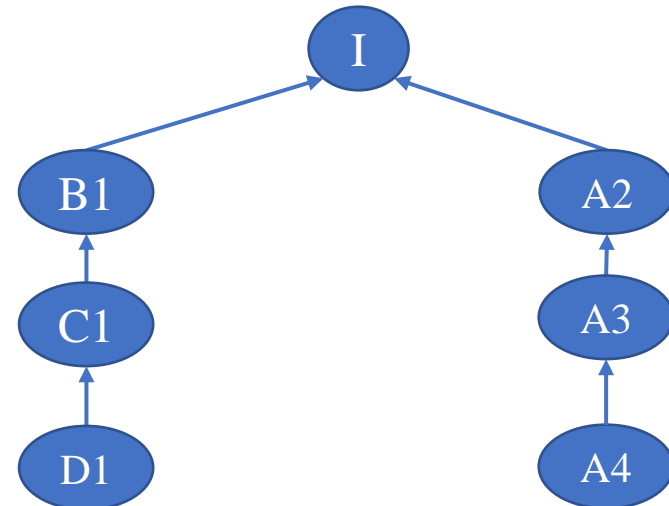
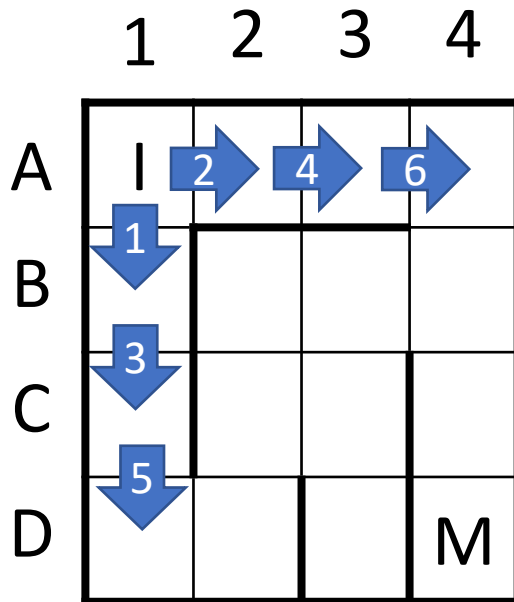
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



Operadores (en orden de prioridad):

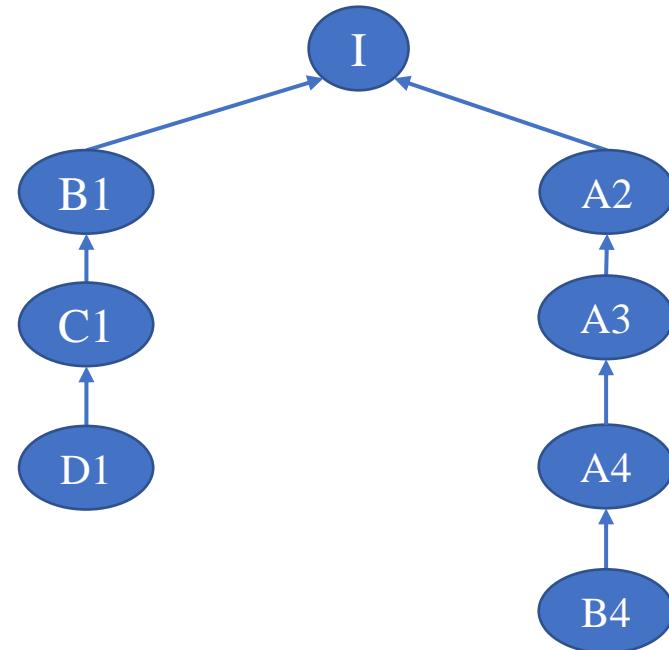
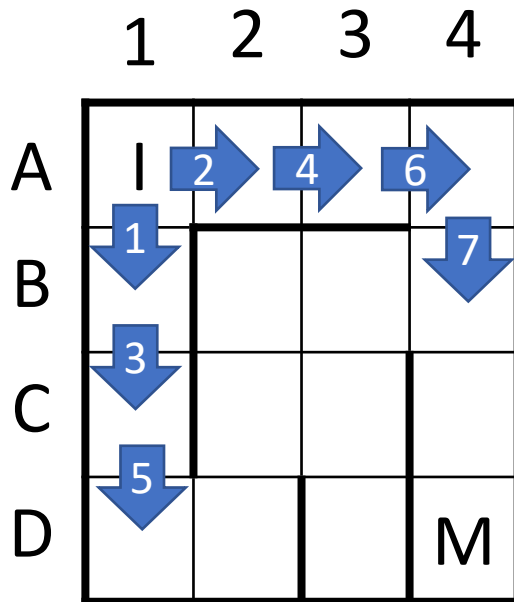
1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha





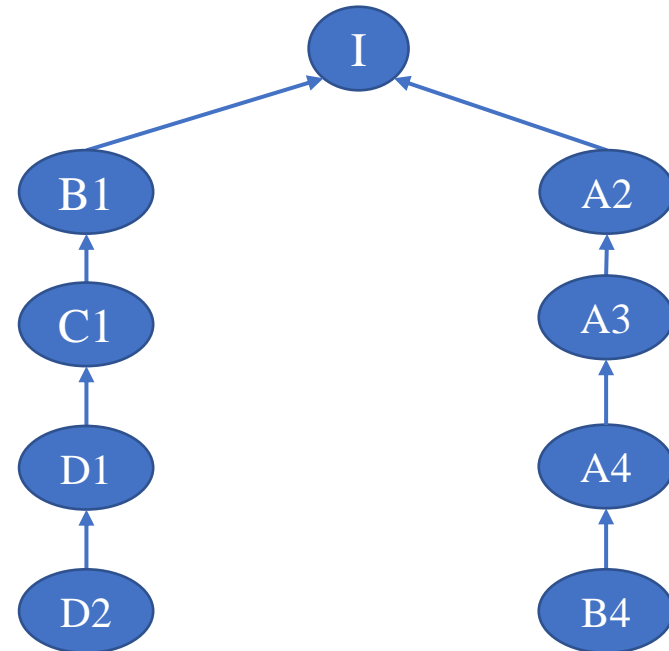
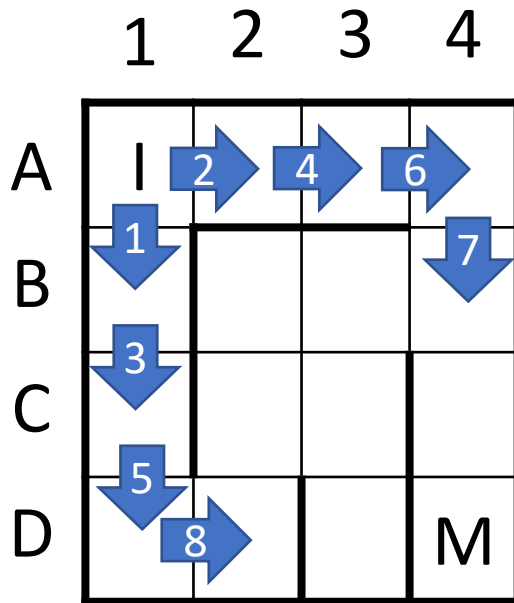
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



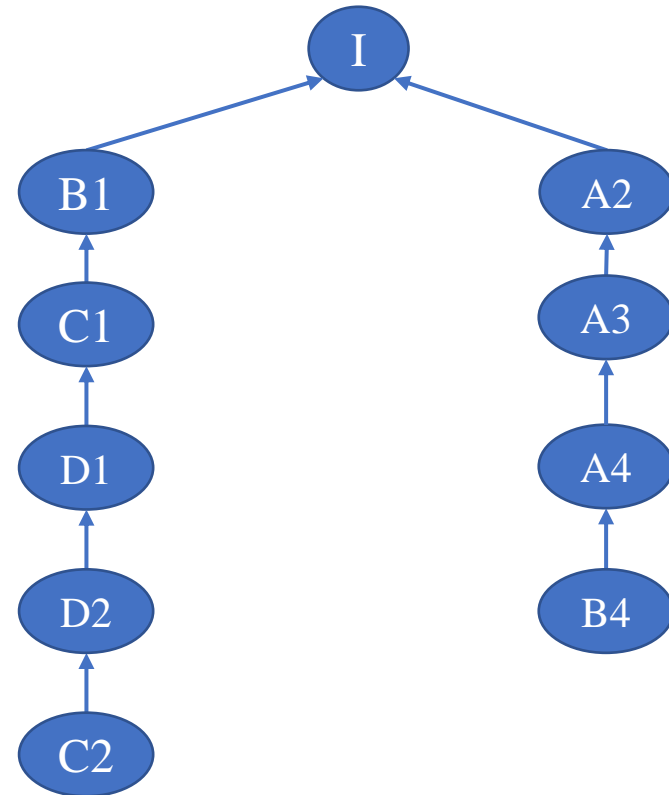
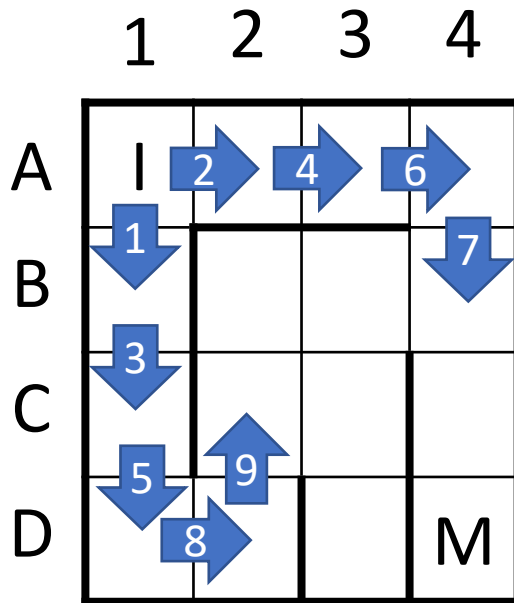
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



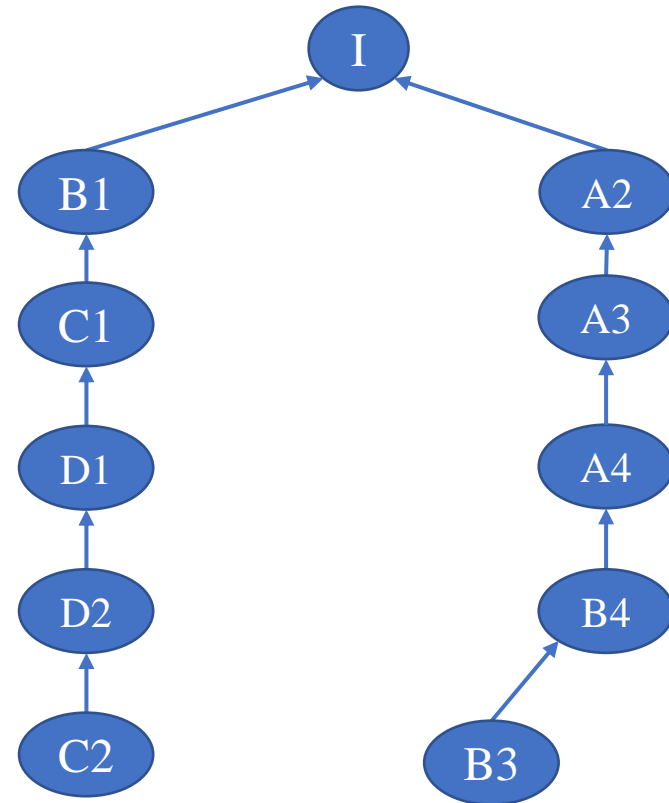
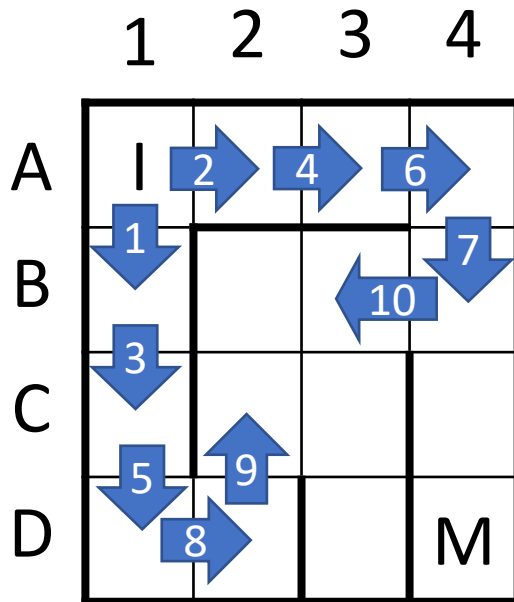
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



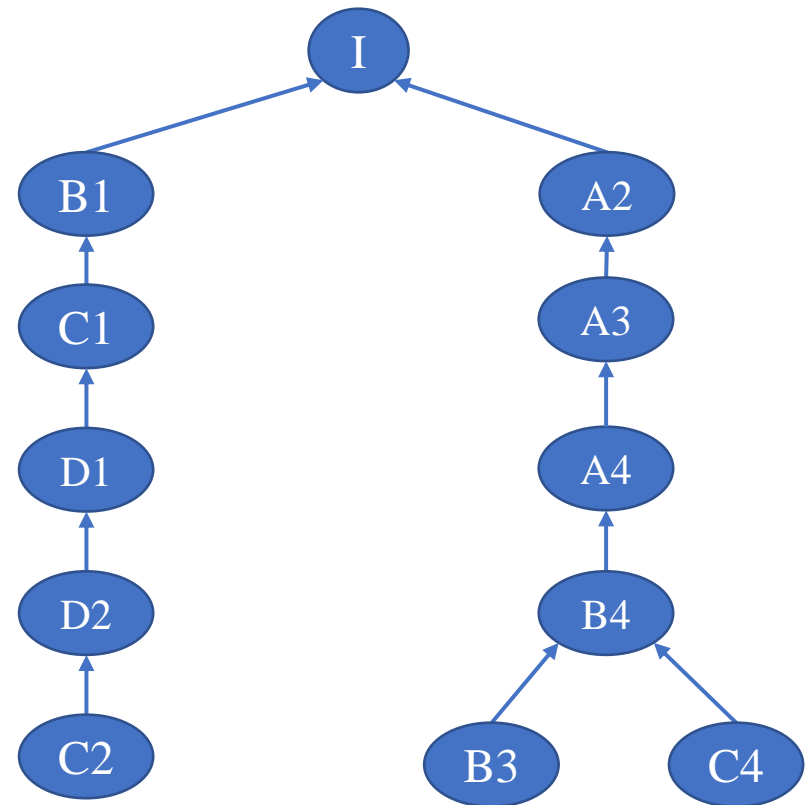
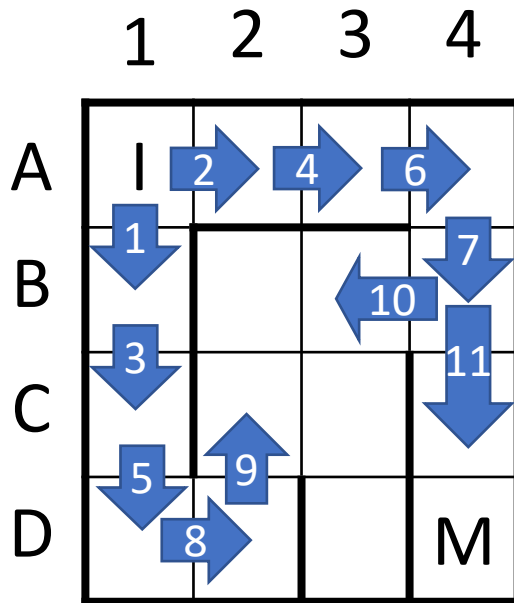
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



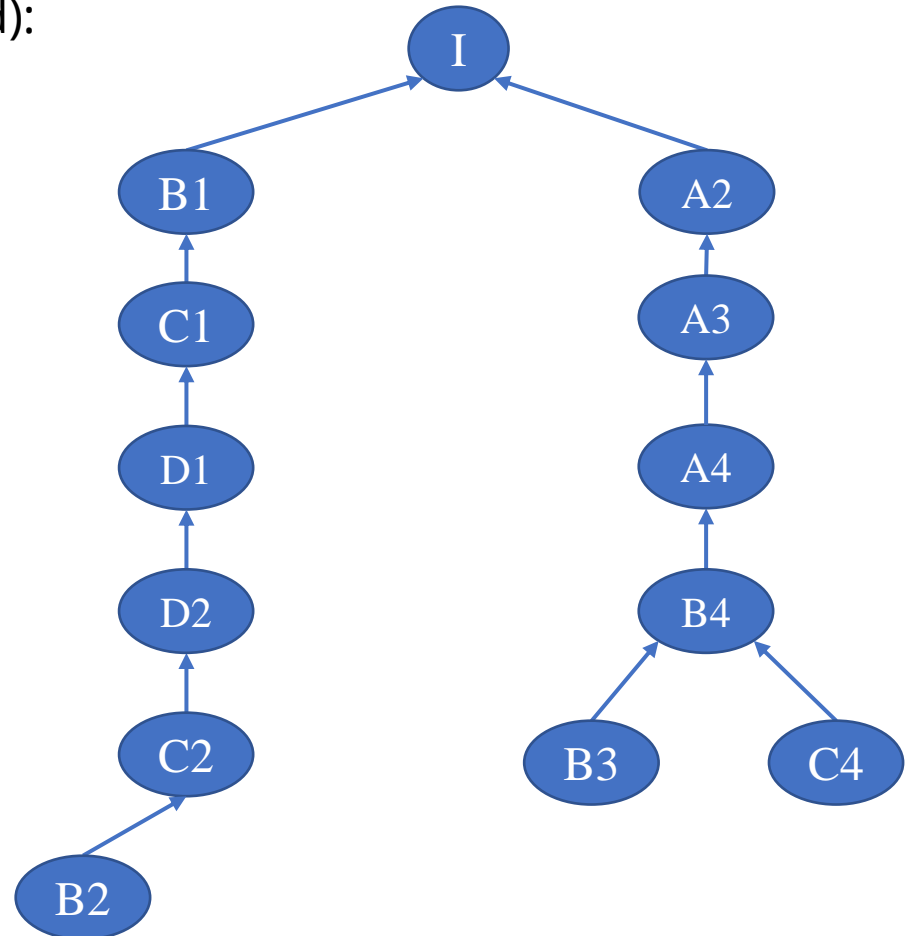
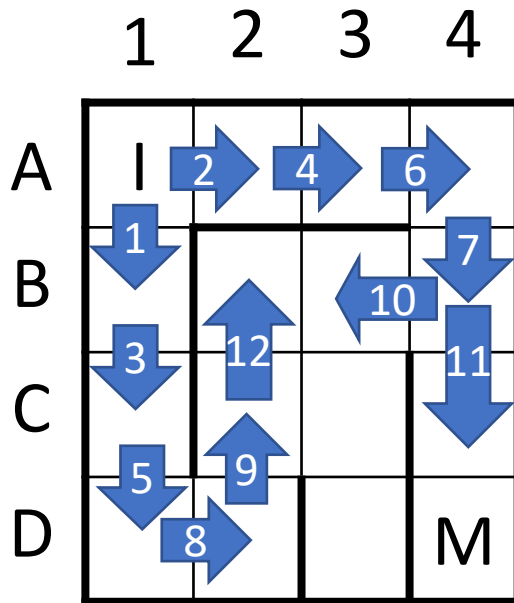
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



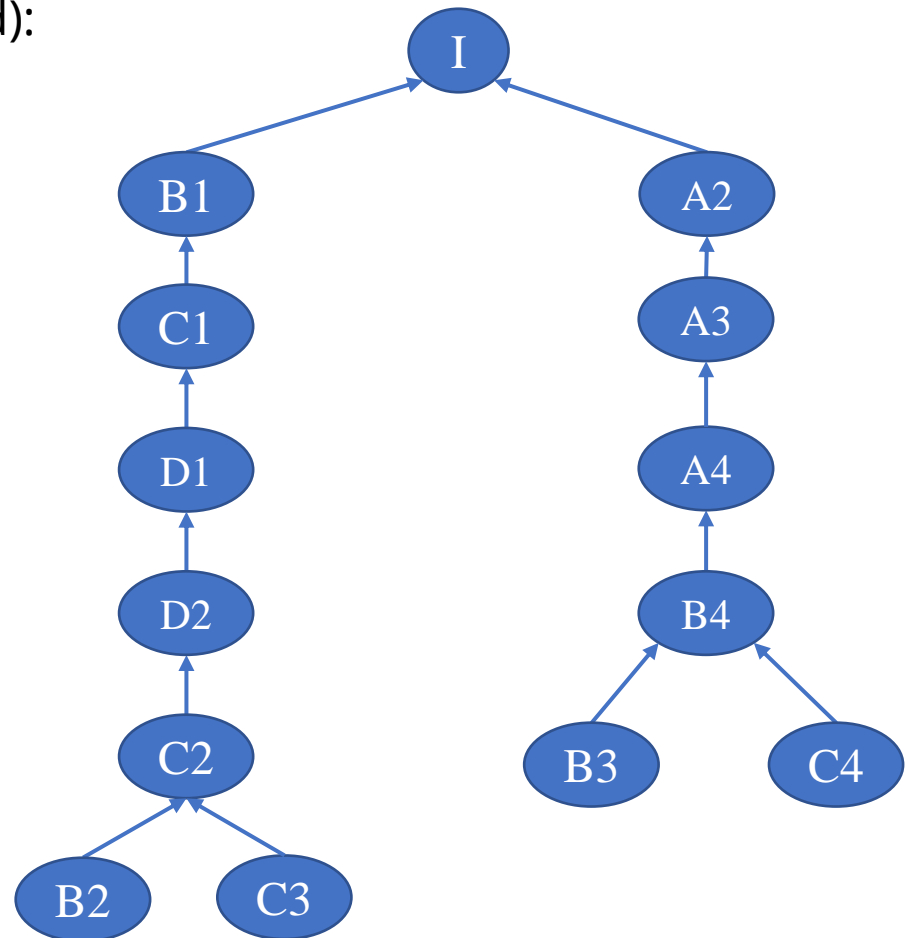
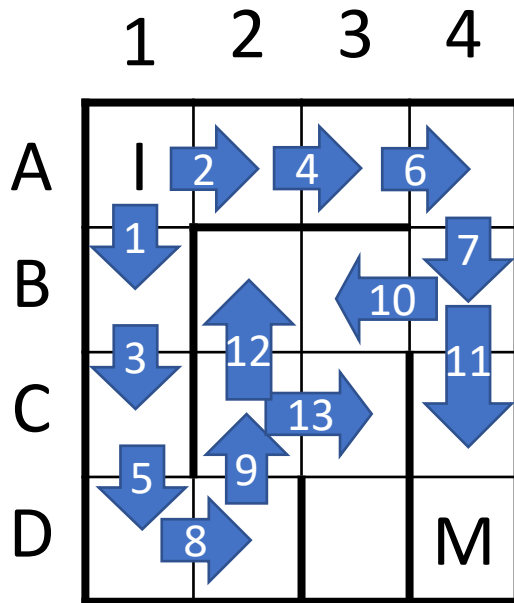
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



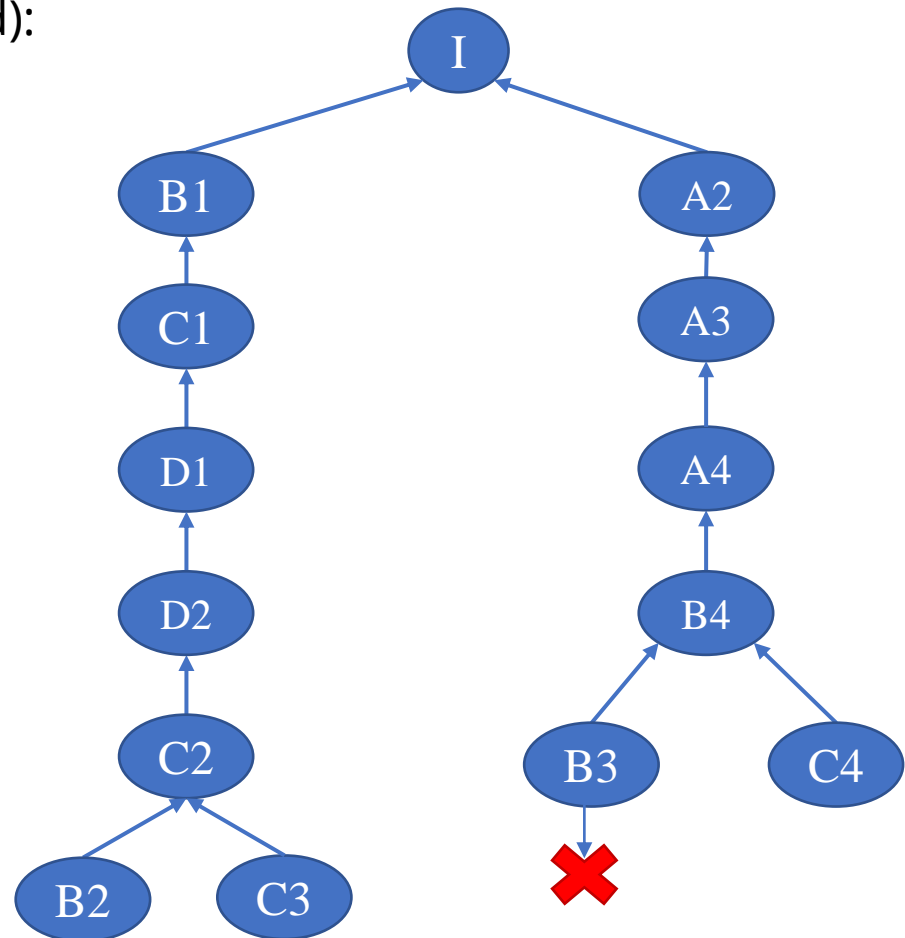
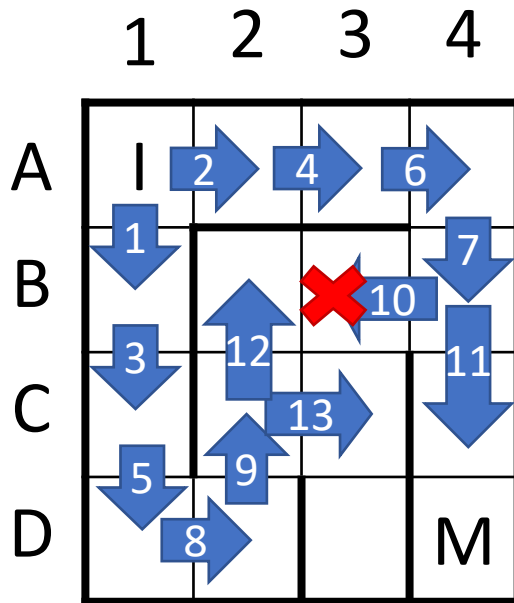
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



Operadores (en orden de prioridad):

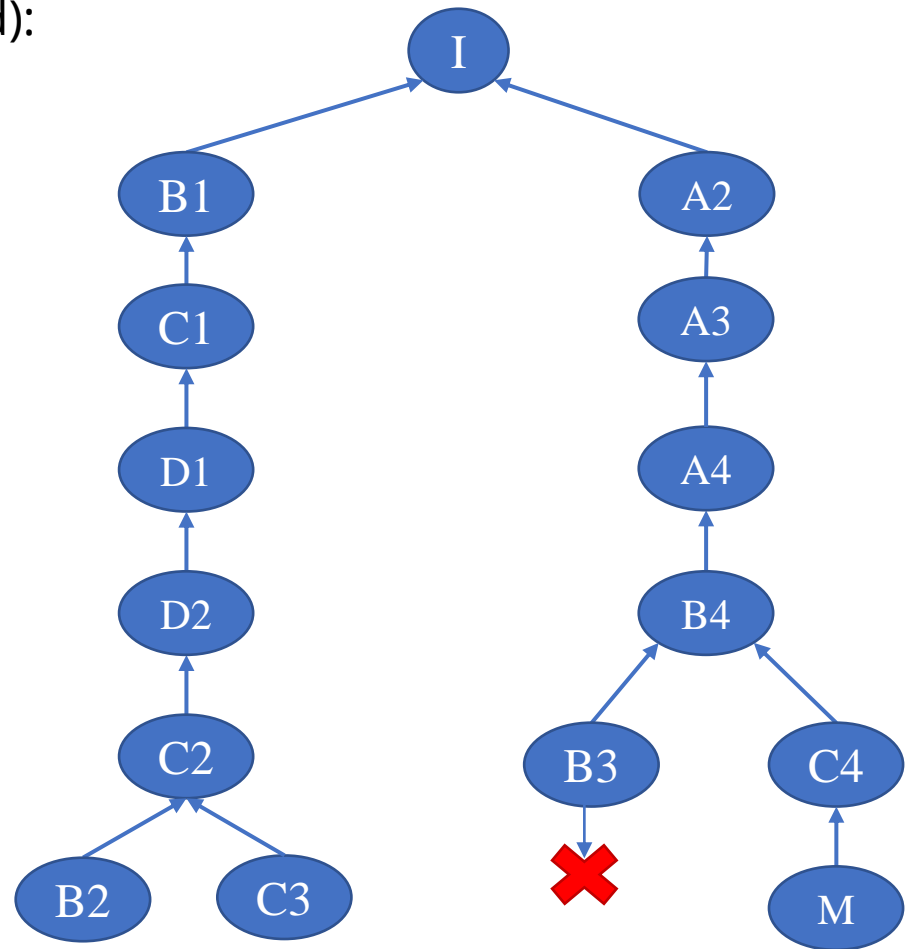
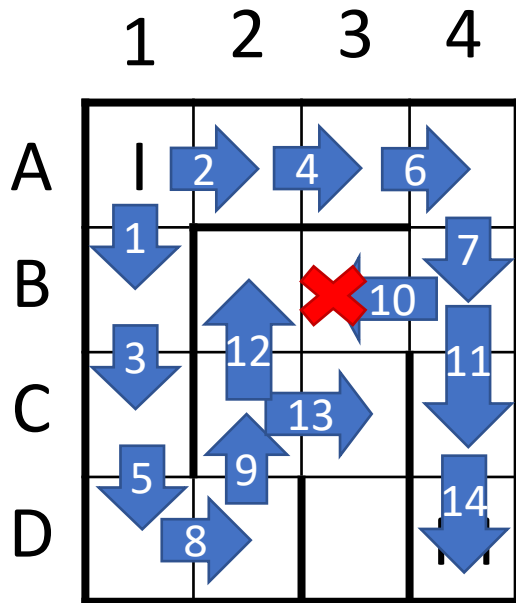
1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha





Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



Operadores (en orden de prioridad):

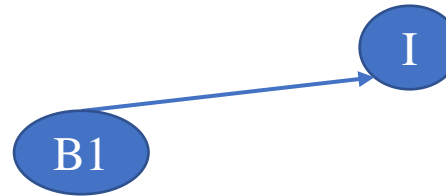
1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha

I

	1	2	3	4
A	I			
B				
C				
D				M

Operadores (en orden de prioridad):

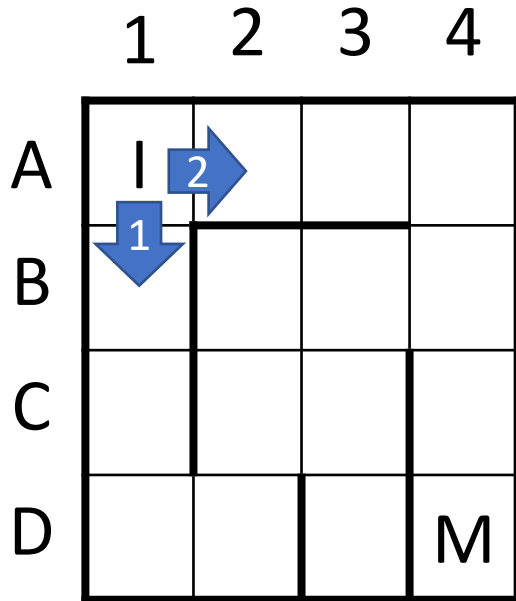
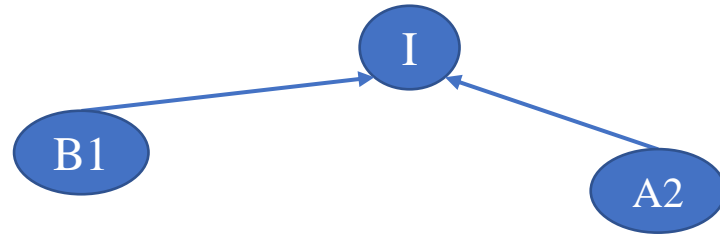
1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



	1	2	3	4
A	I ↓ 1			
B				
C				
D				M

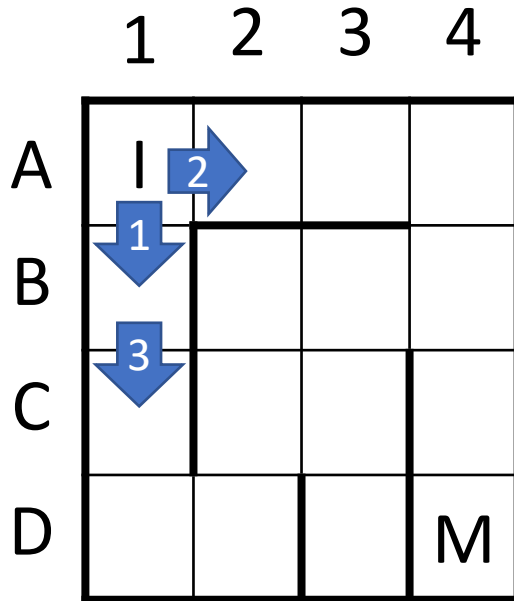
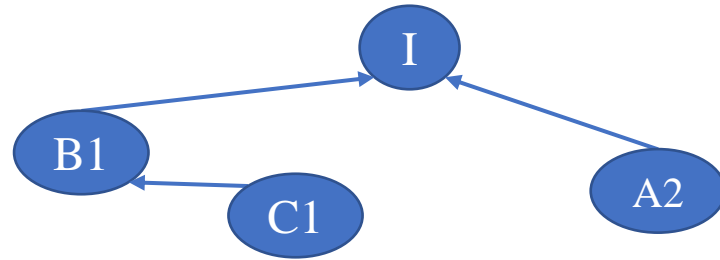
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



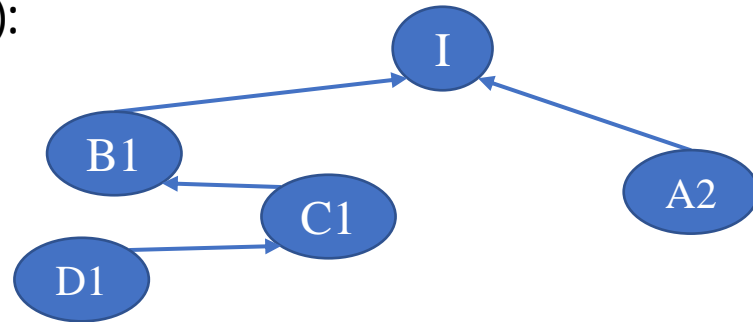
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



Operadores (en orden de prioridad):

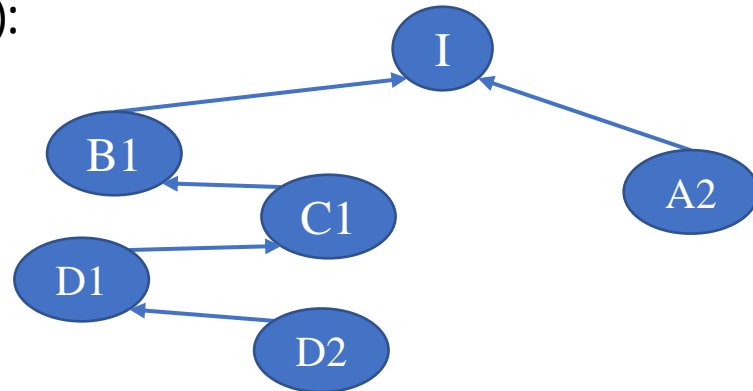
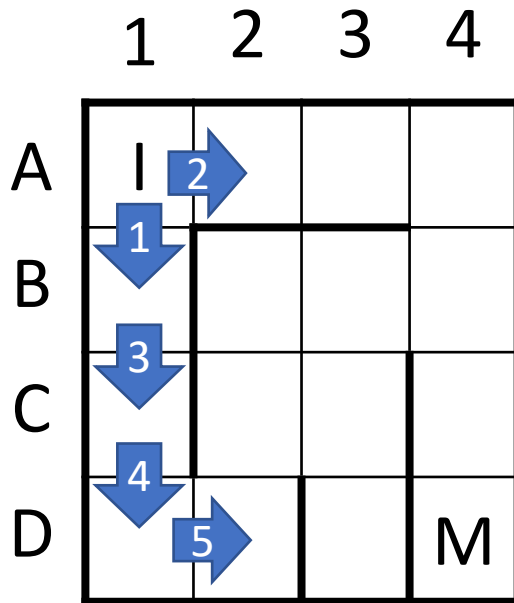
1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



	1	2	3	4
A	I			
B				
C				
D				M

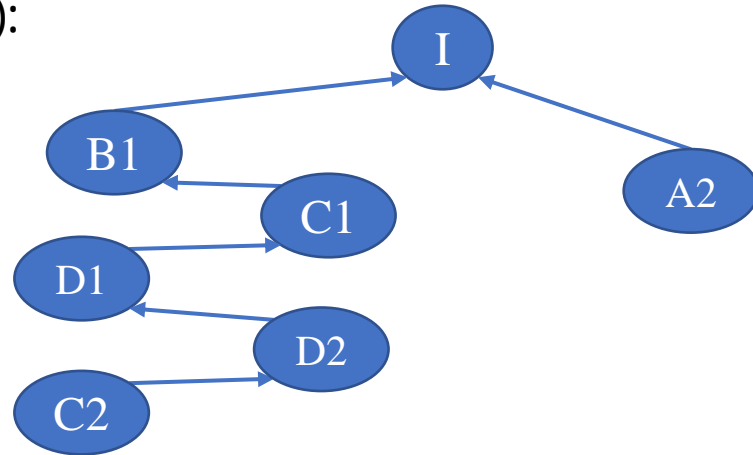
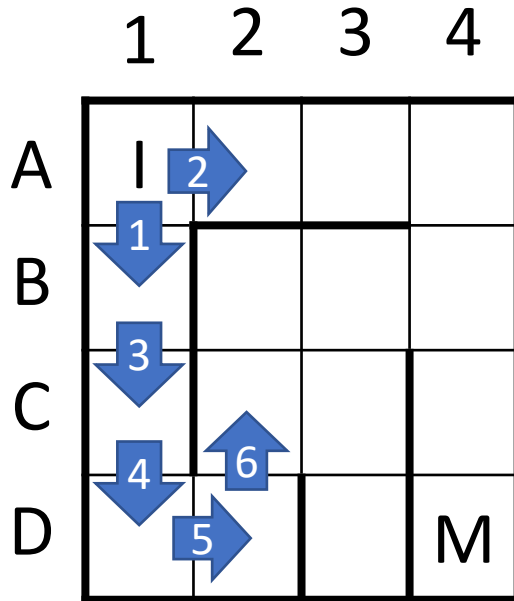
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



Operadores (en orden de prioridad):

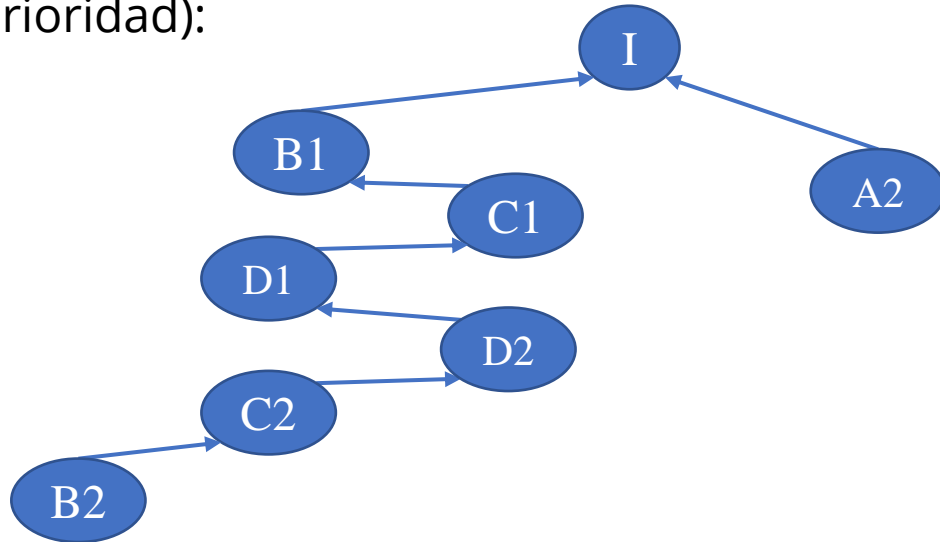
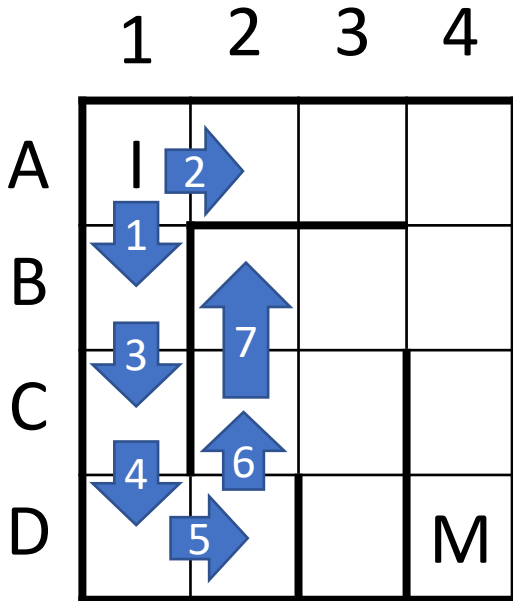
1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha





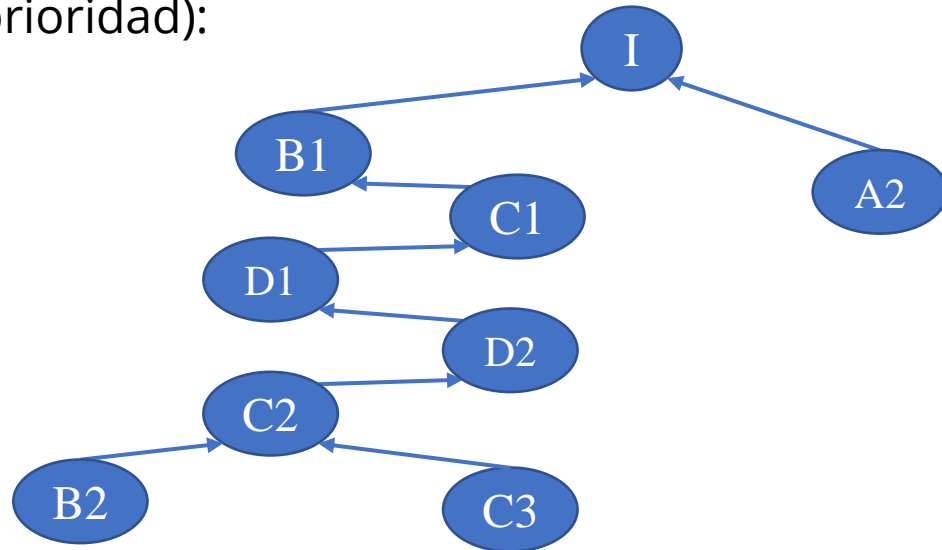
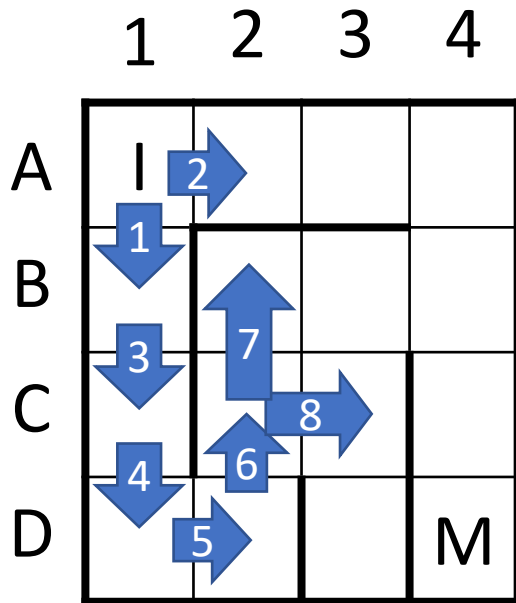
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



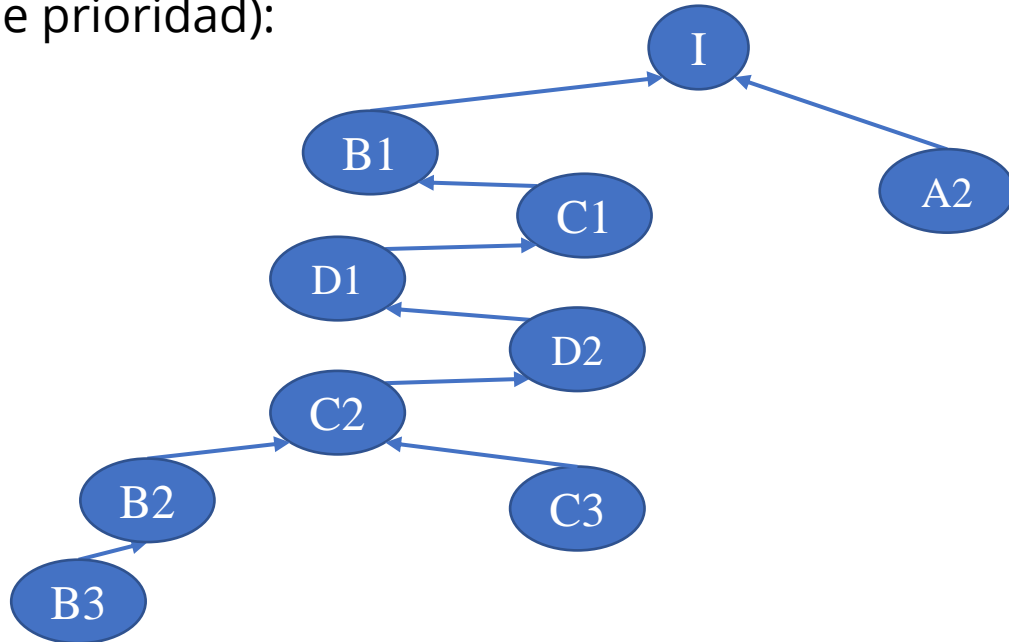
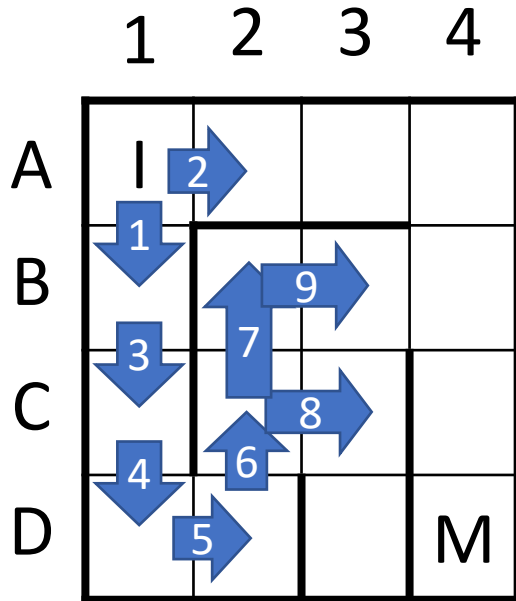
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



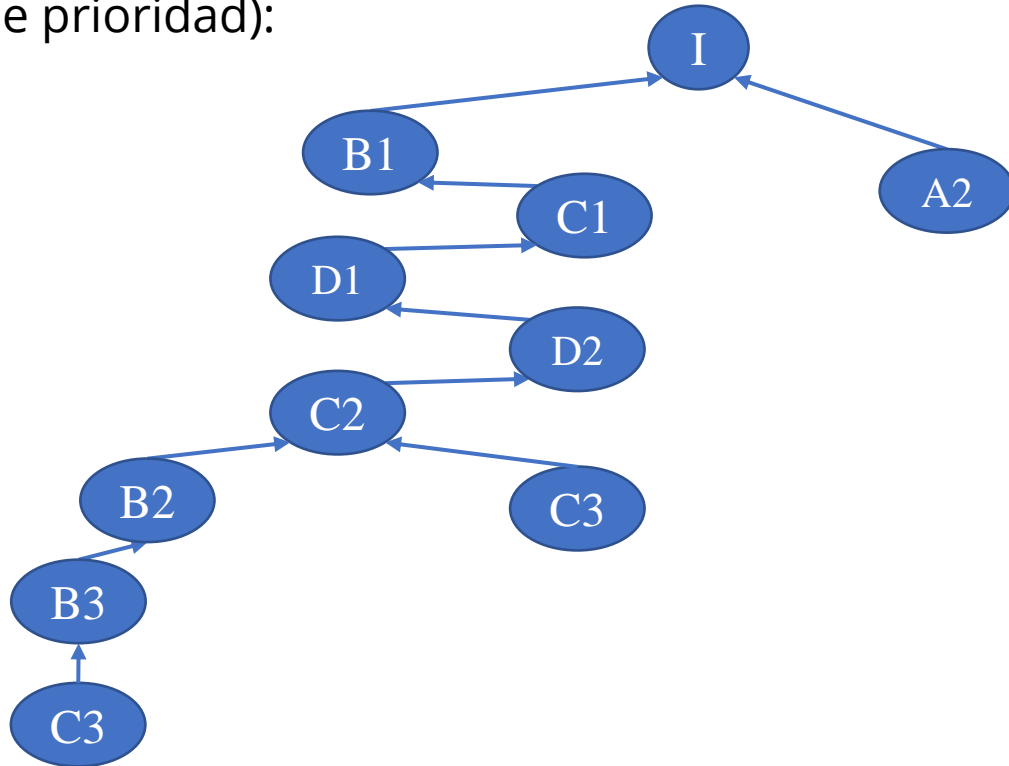
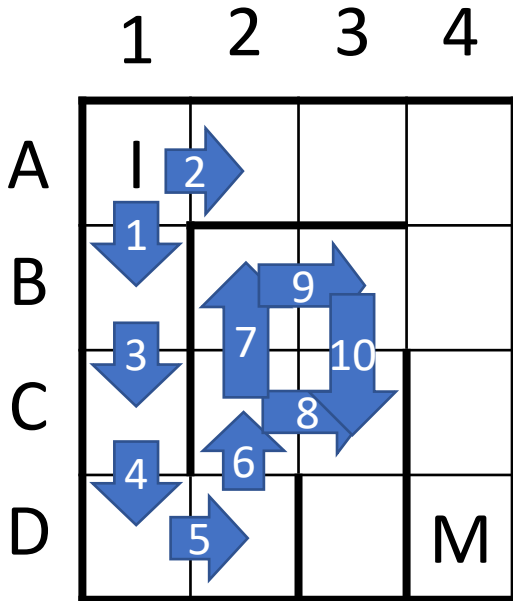
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



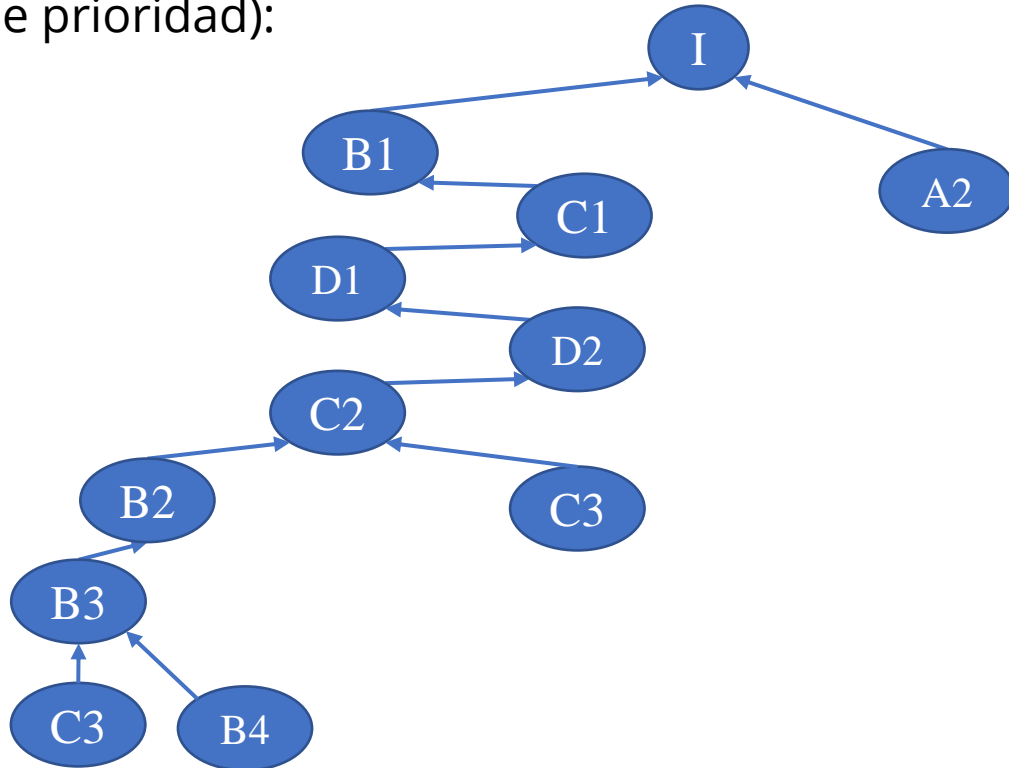
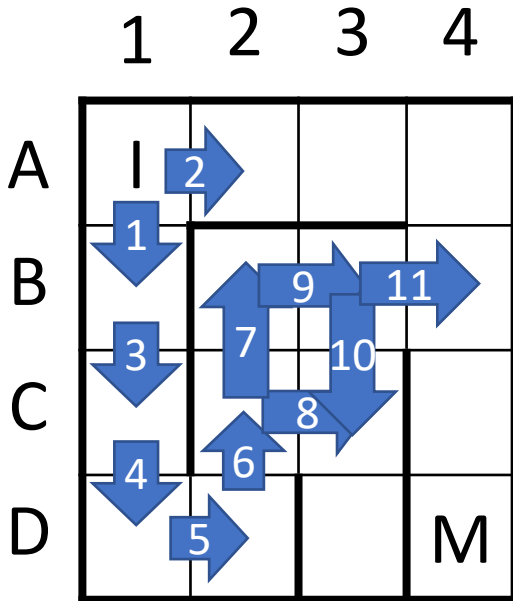
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



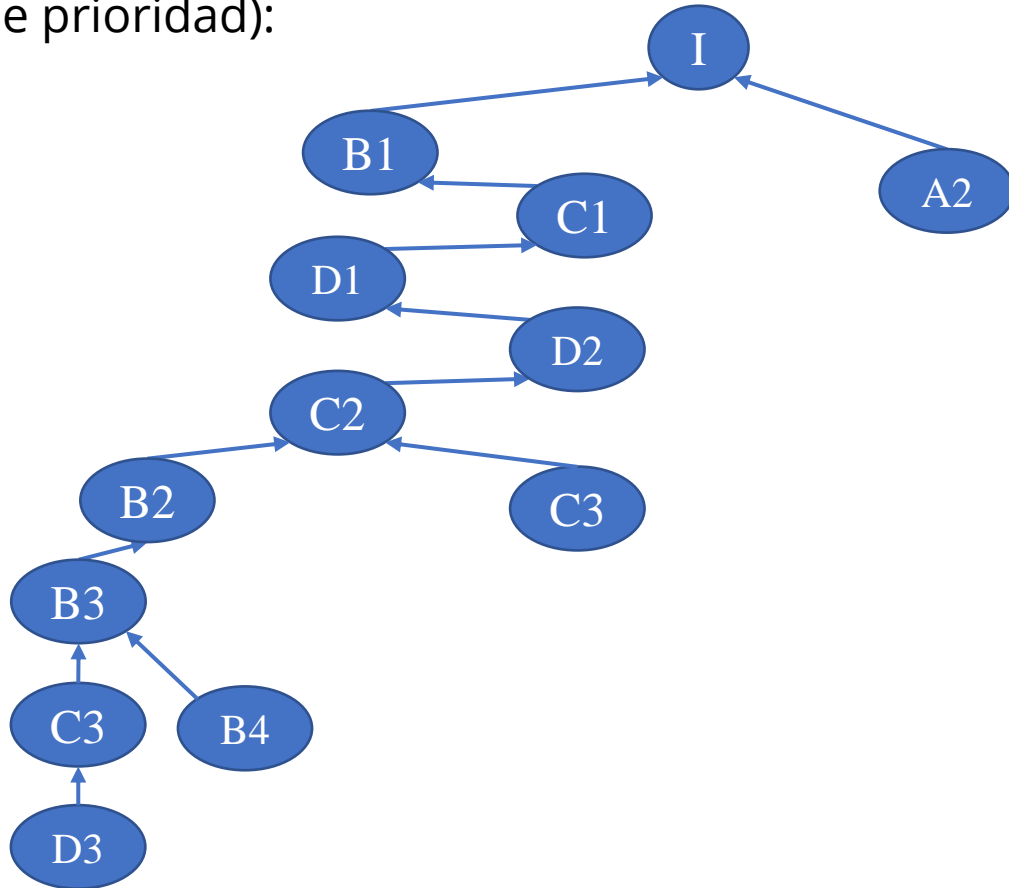
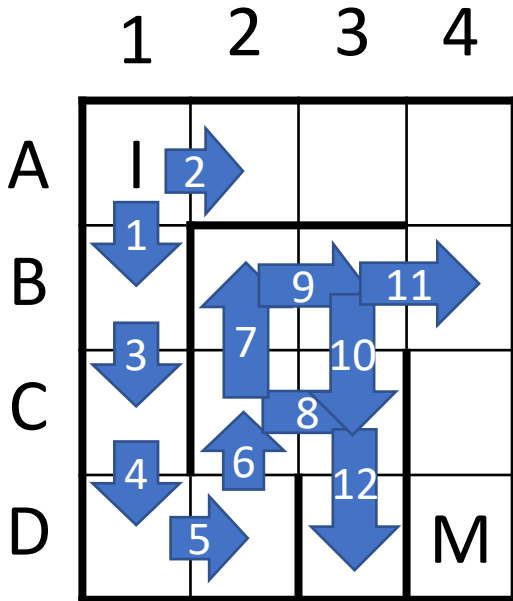
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



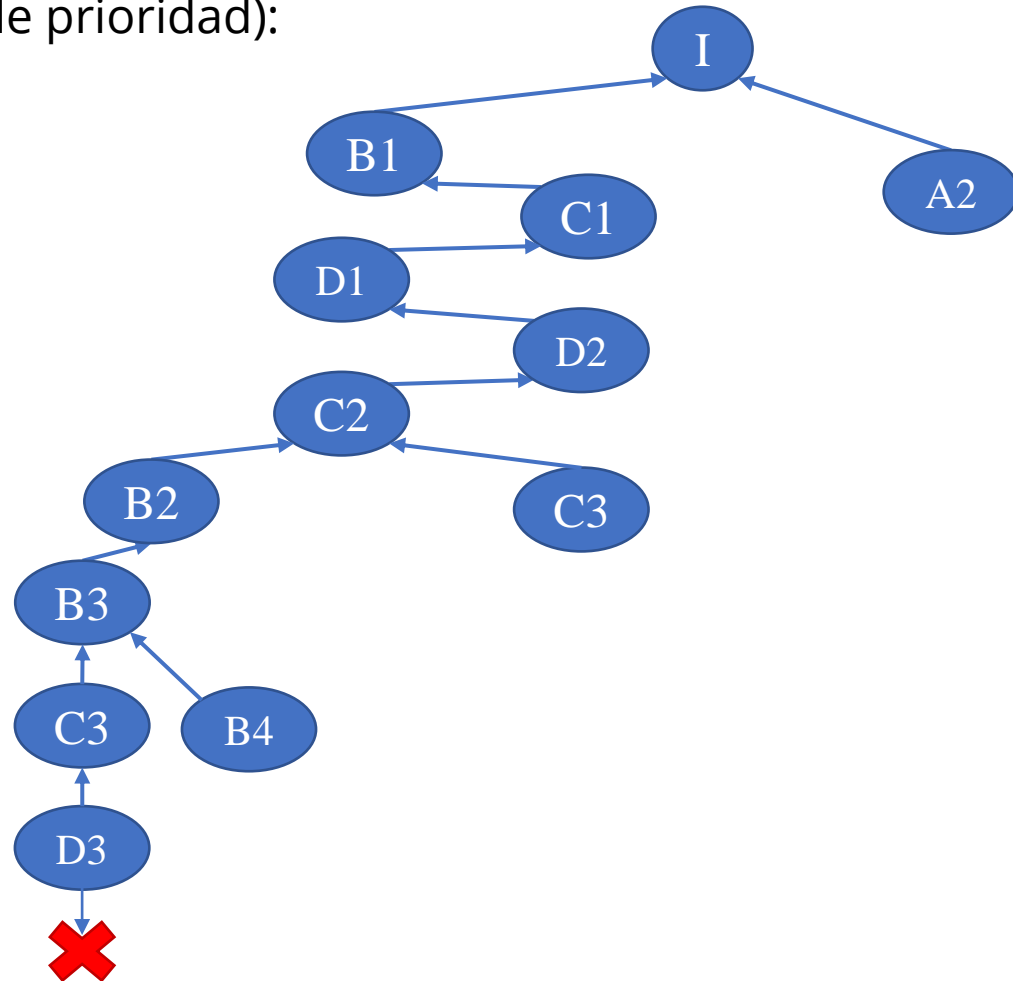
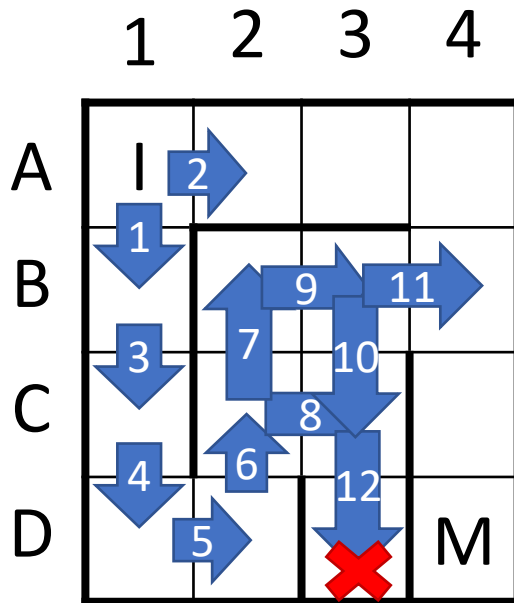
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



Operadores (en orden de prioridad):

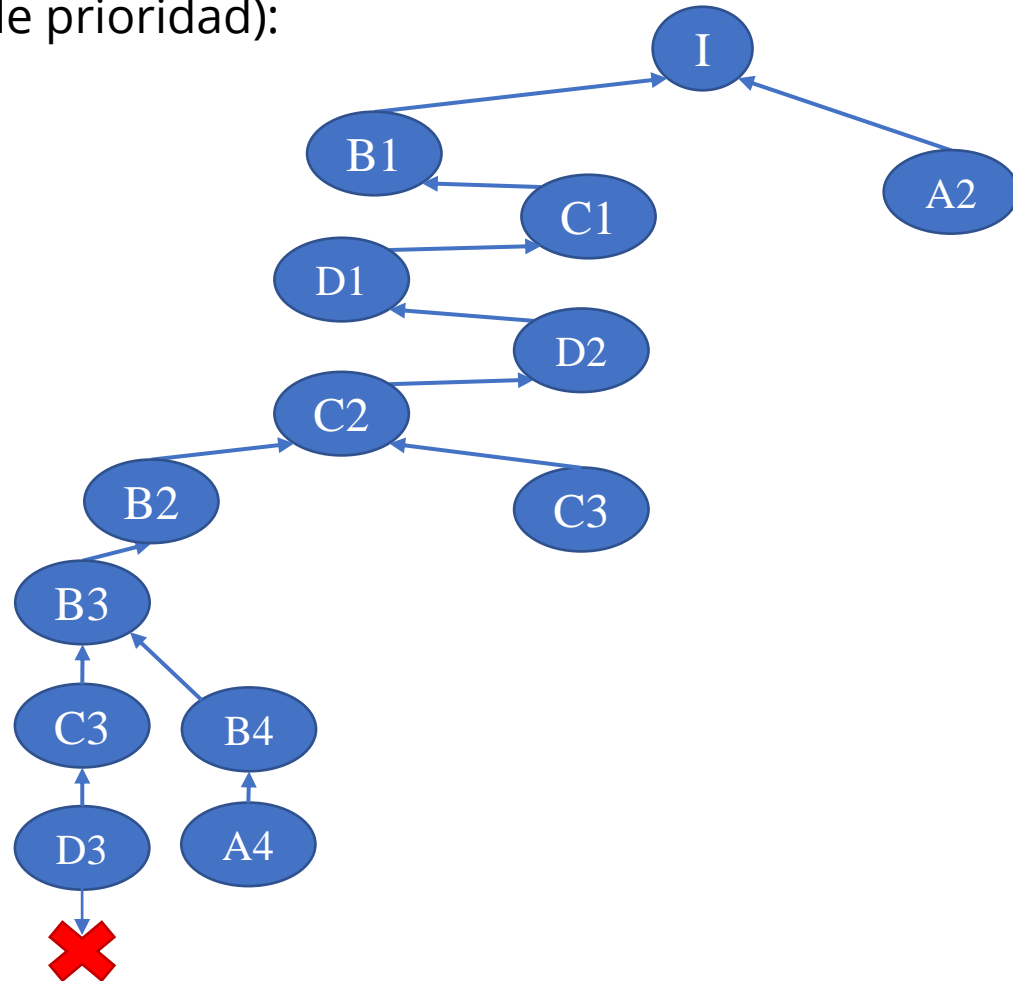
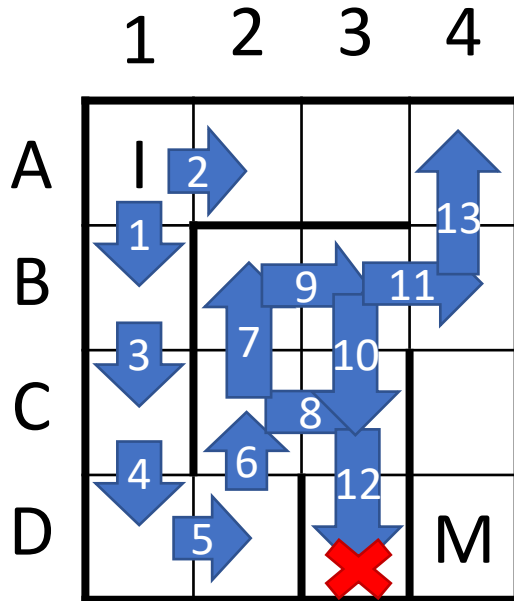
1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



## Solución en profundidad

Operadores (en orden de prioridad):

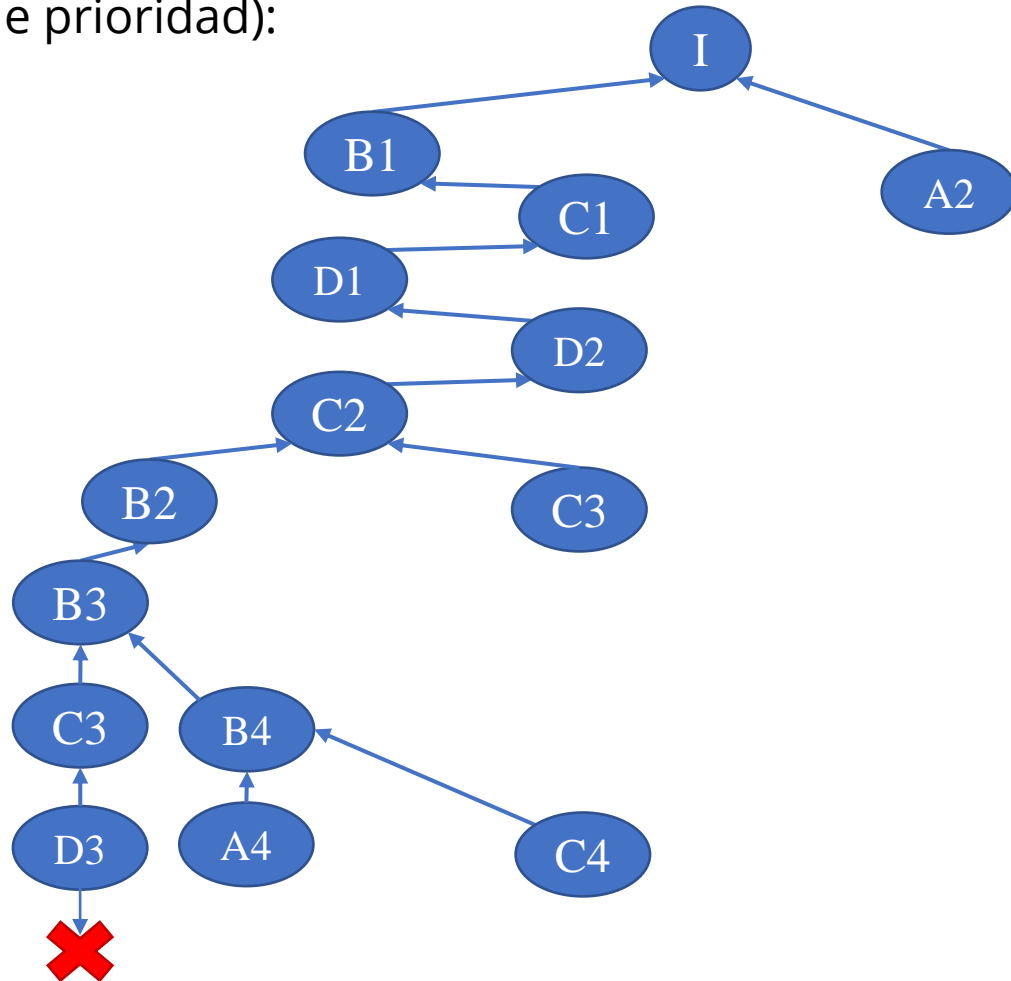
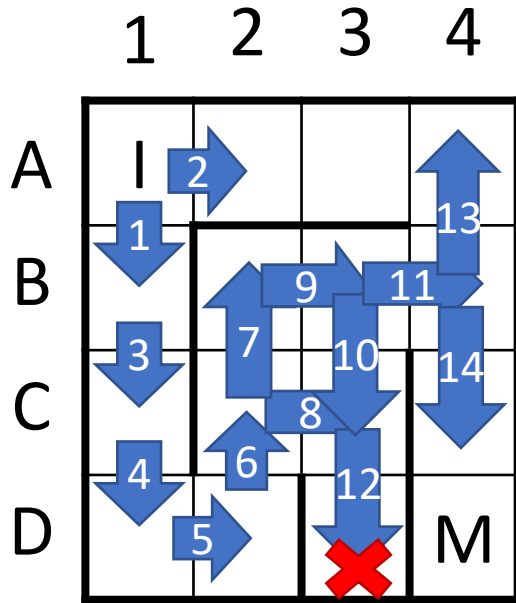
1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha





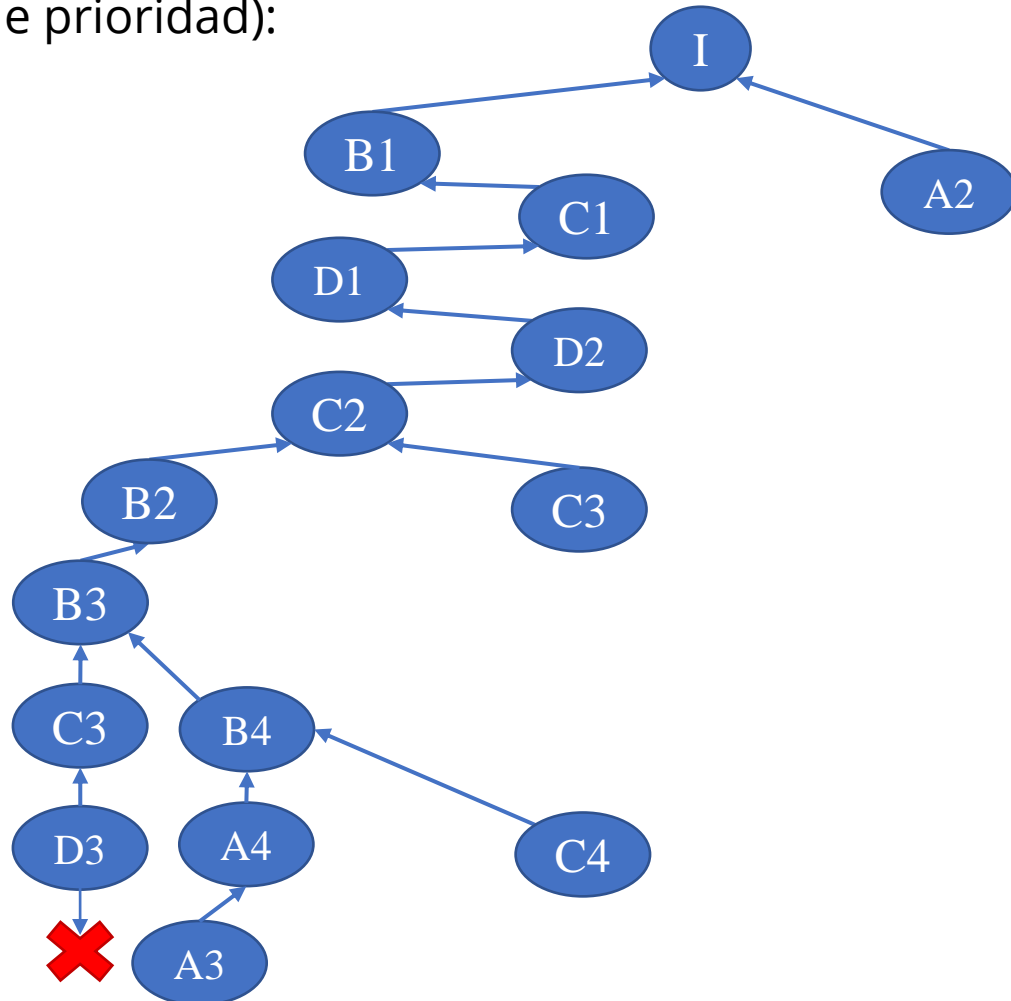
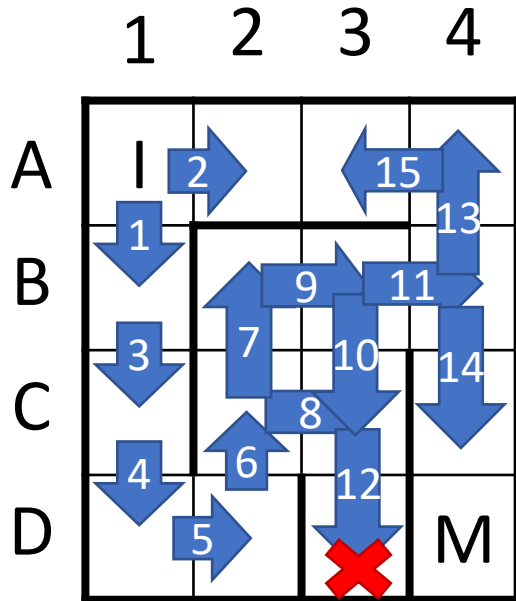
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



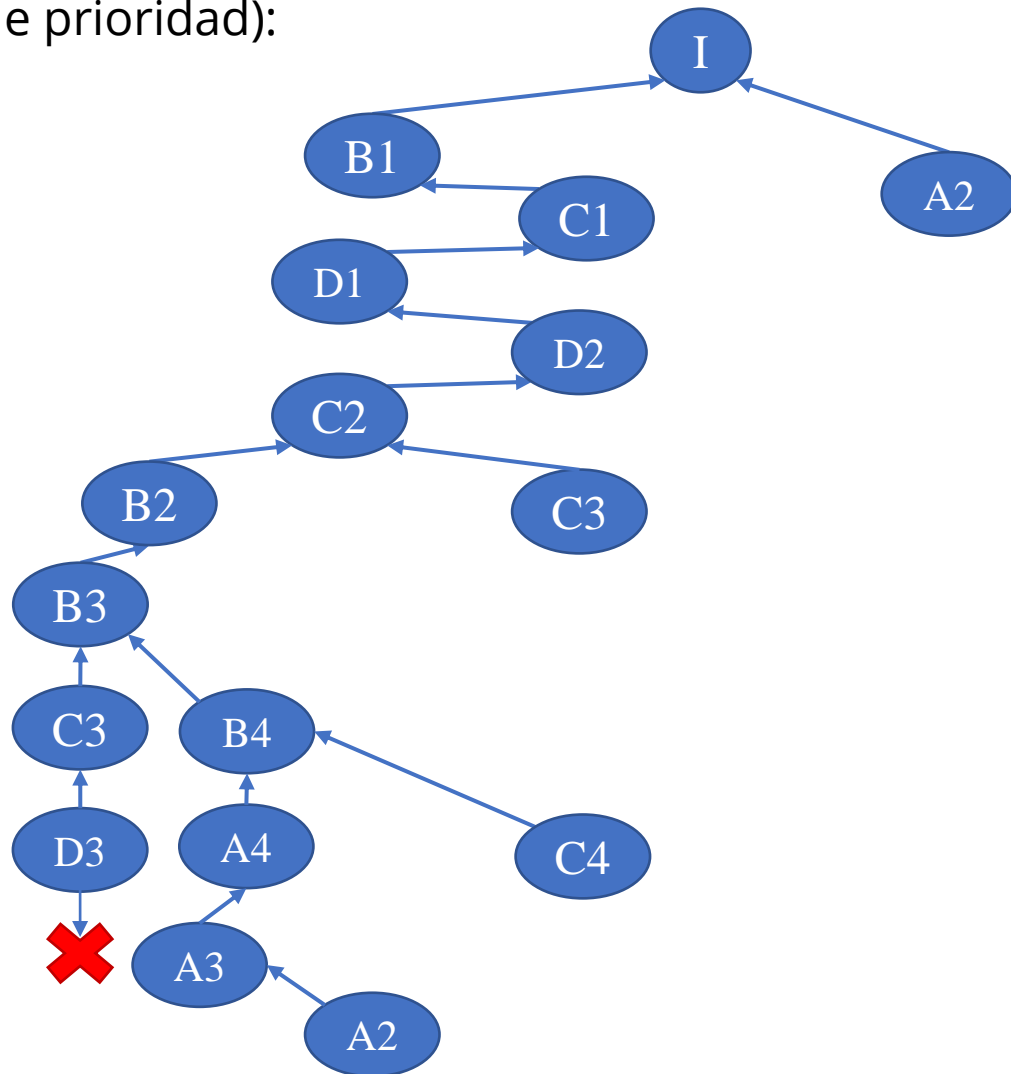
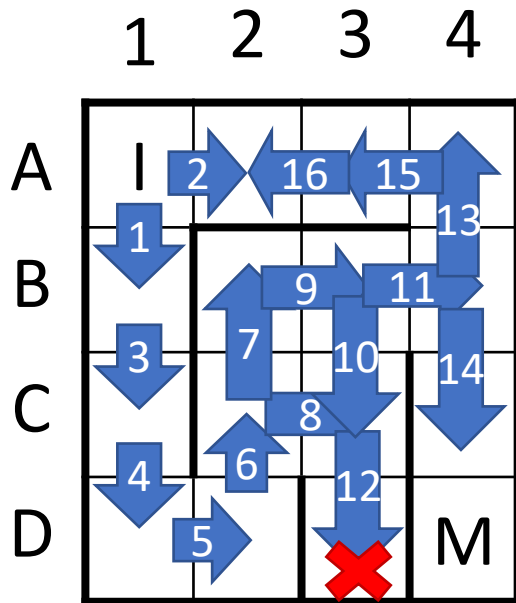
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



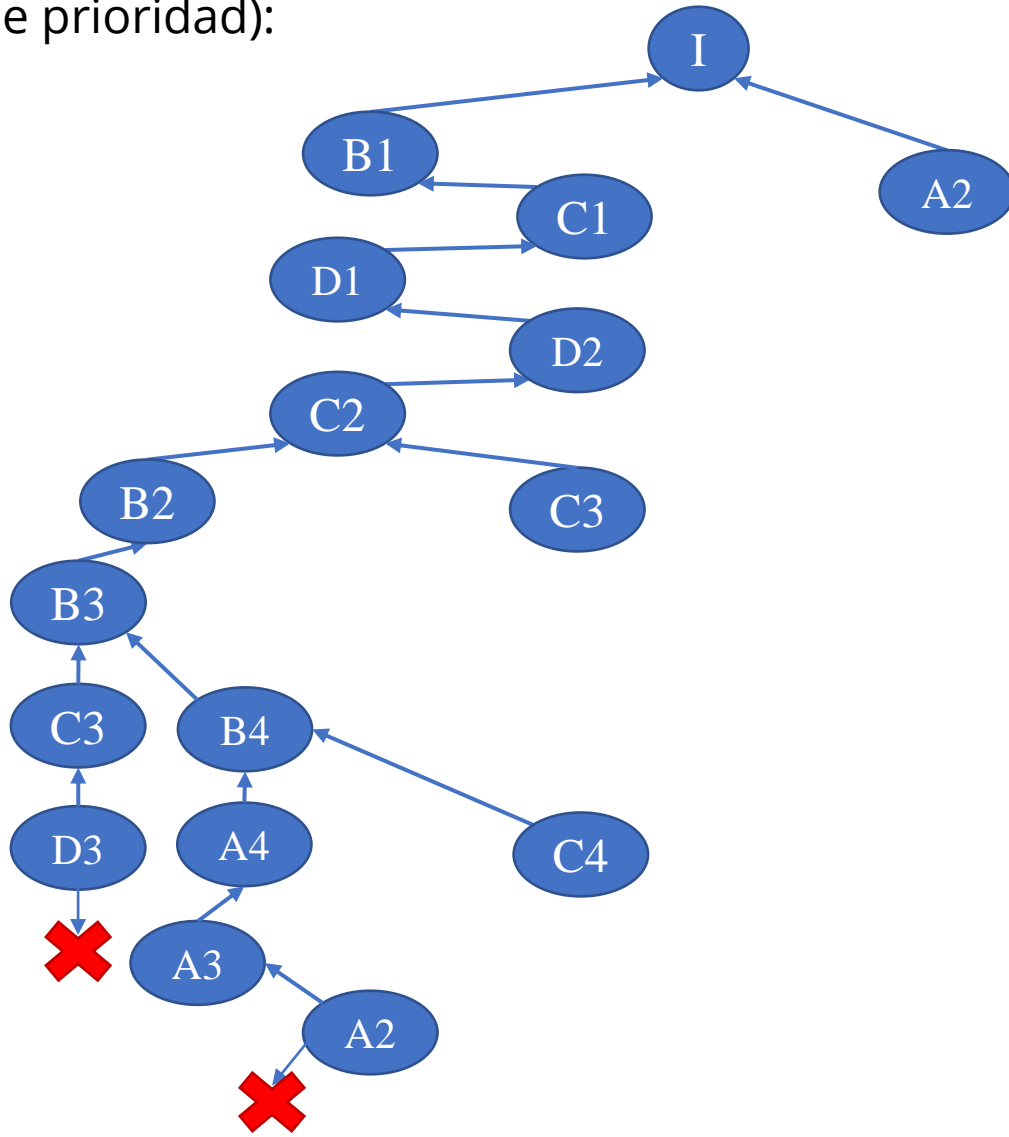
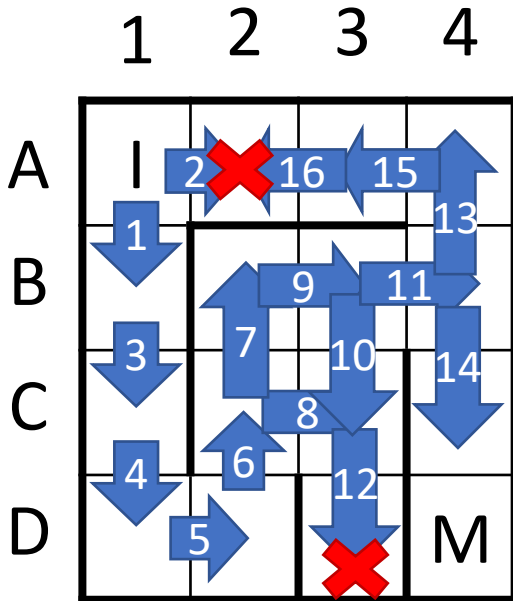
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



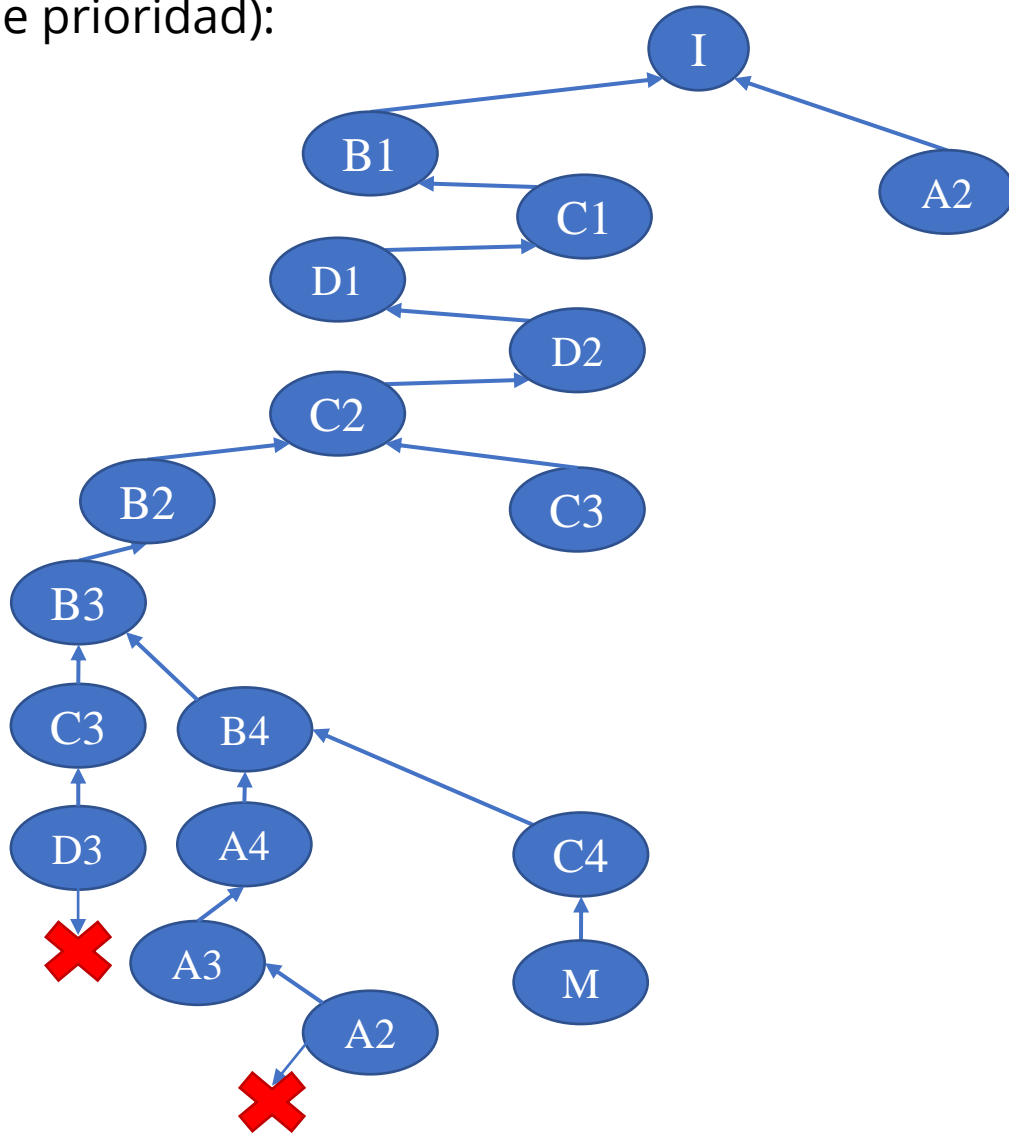
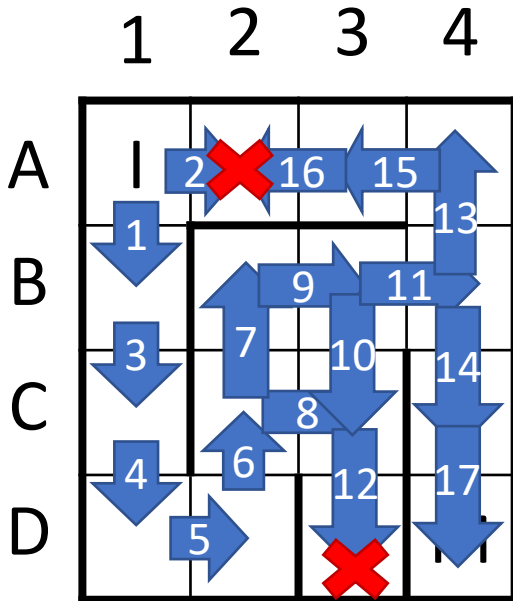
Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



Operadores (en orden de prioridad):

1. Mover arriba
2. Mover izquierda
3. Mover abajo
4. Mover derecha



El TopSpin es un juego de permutaciones que consiste en una ruleta de números que puede girar y una base giratoria que permite intercambiar dos números adyacentes.

El objetivo del juego es ordenar los números colocando la base giratoria en el par (1,2)

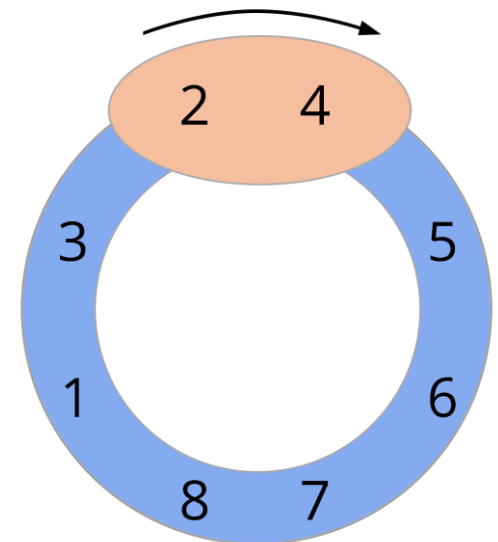
Los operadores serían (en orden de aplicación):

1. Mover números a la derecha
2. Mover números a la izquierda
3. Rotar base giratoria

Partiendo del estado inicial de la imagen mostrada.

### Preguntas:

- Hacer la búsqueda en amplitud
- Hacer búsqueda en profundidad con profundidad máxima de 4



**Representar el estado**

**Representar los operadores**

## **Representar el estado**

Para representar el estado podemos dar la cadena de números en sentido horario empezando por el primer número de la base giratoria.

Por ejemplo el estado meta quedaría así: 12 - 345678

## **Representar los operadores**



## Representar el estado

Para representar el estado podemos dar la cadena de números en sentido horario empezando por el primer número de la base giratoria.

Por ejemplo el estado meta quedaría así: **12 - 345678**

## Representar los operadores

Los operadores quedarán de la siguiente forma:

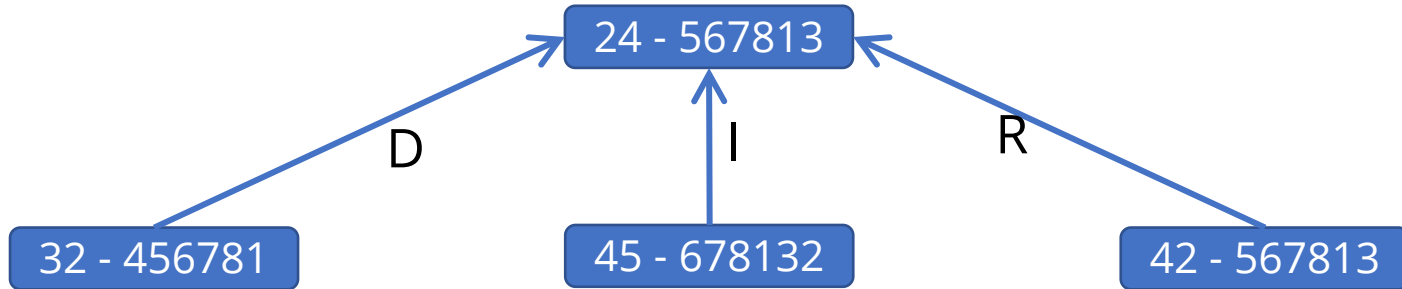
1. Mover números a la derecha → D **12 - 345678** → **81 - 234567**
2. Mover números a la izquierda → I **12 - 345678** → **23 - 456781**
3. Rotar base giratoria → R **12 - 345678** → **21 - 345678**

# Solución en amplitud

24 - 567813

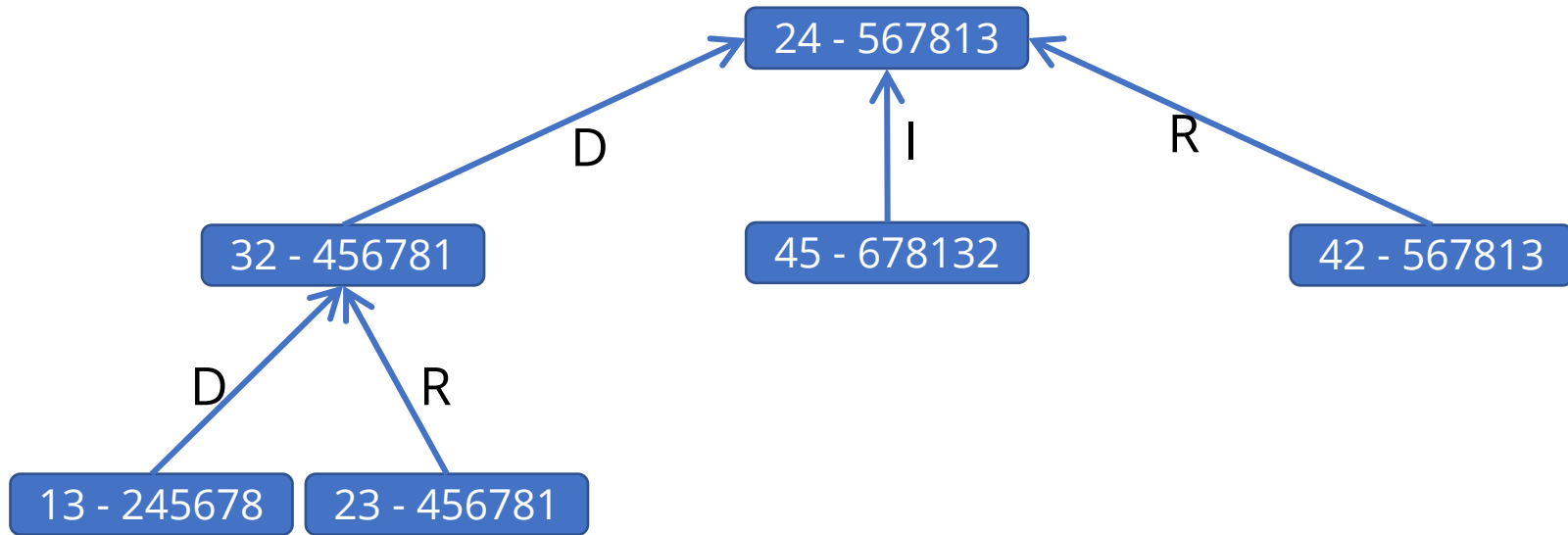
# Ejercicio 5

## Solución en amplitud



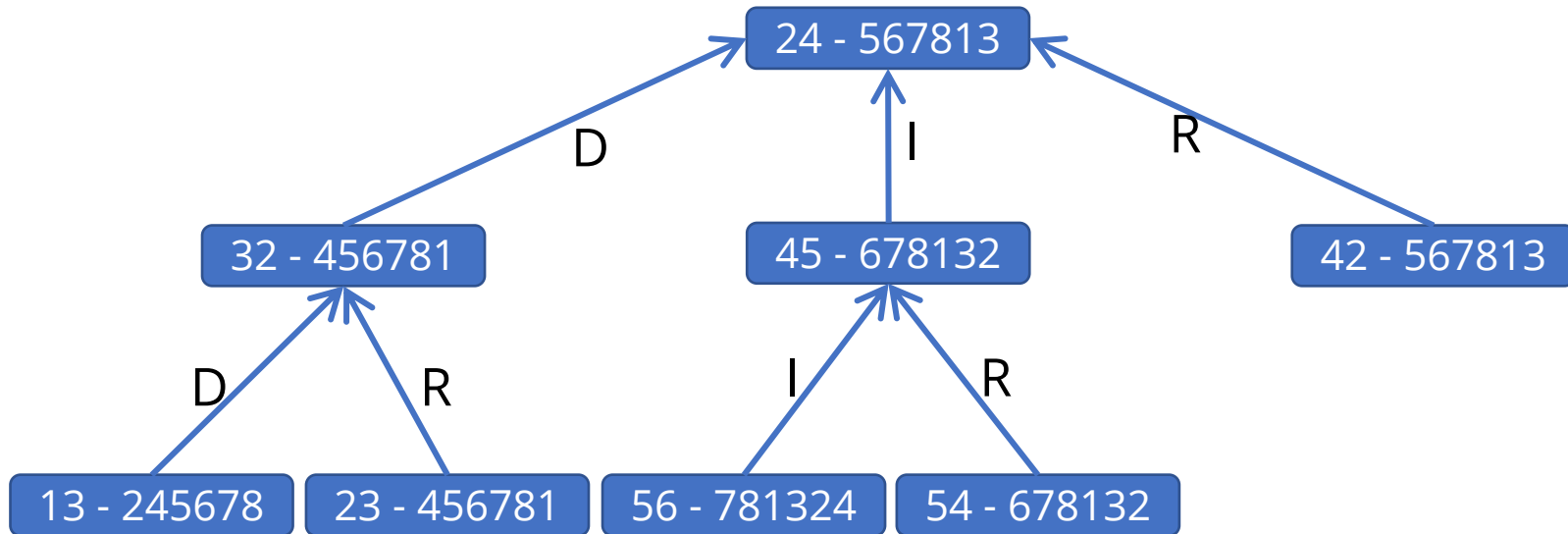
# Ejercicio 5

## Solución en amplitud



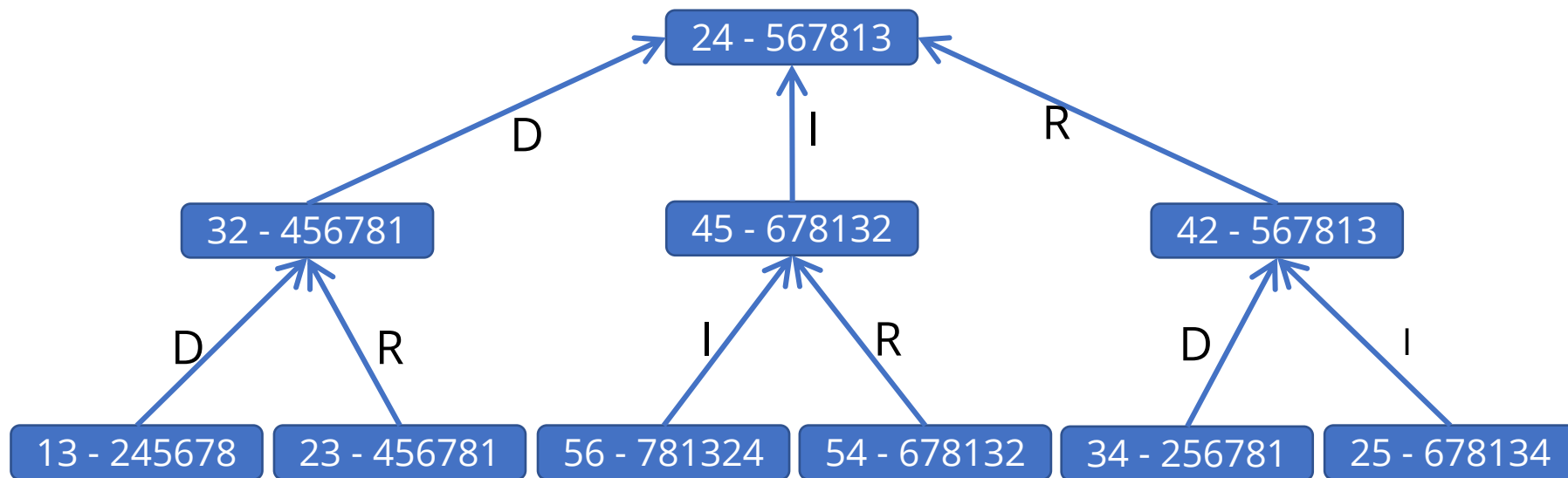
# Ejercicio 5

## Solución en amplitud



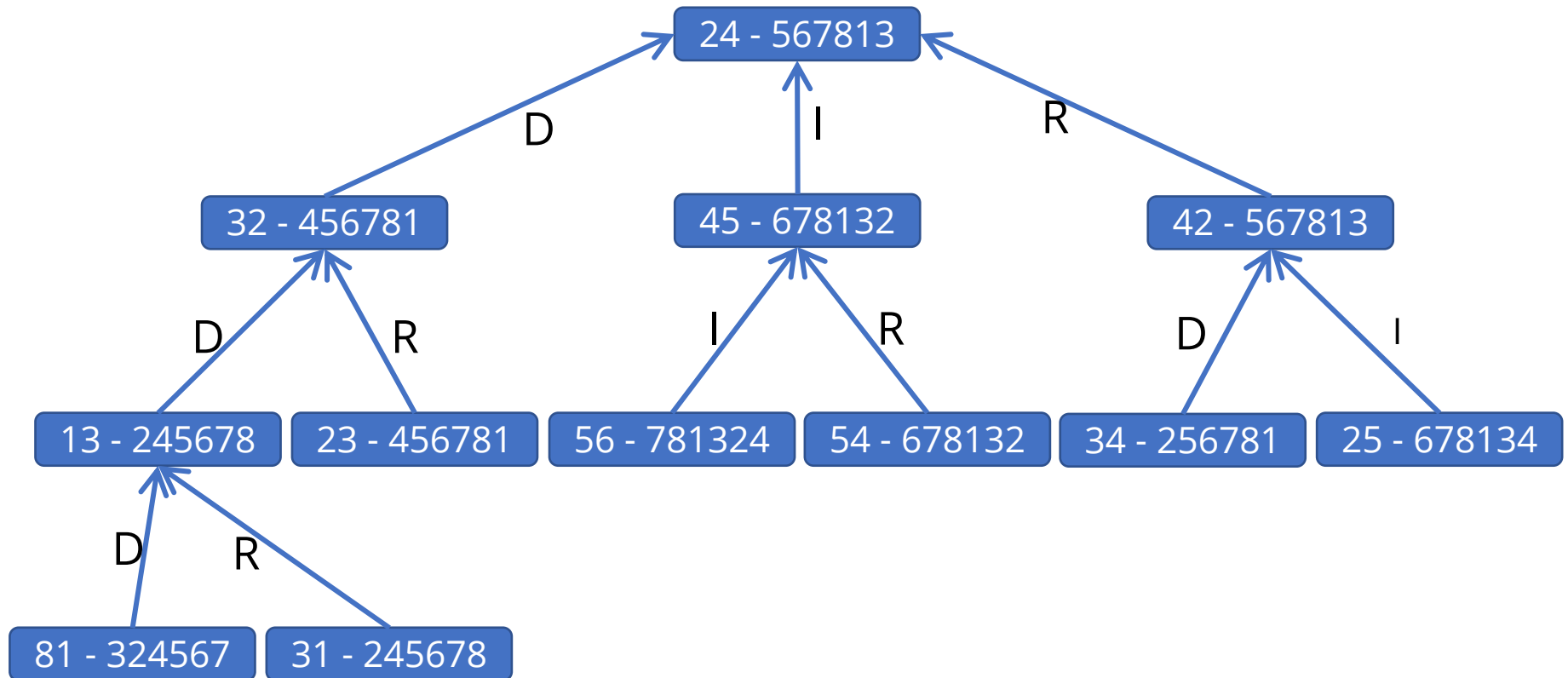
# Ejercicio 5

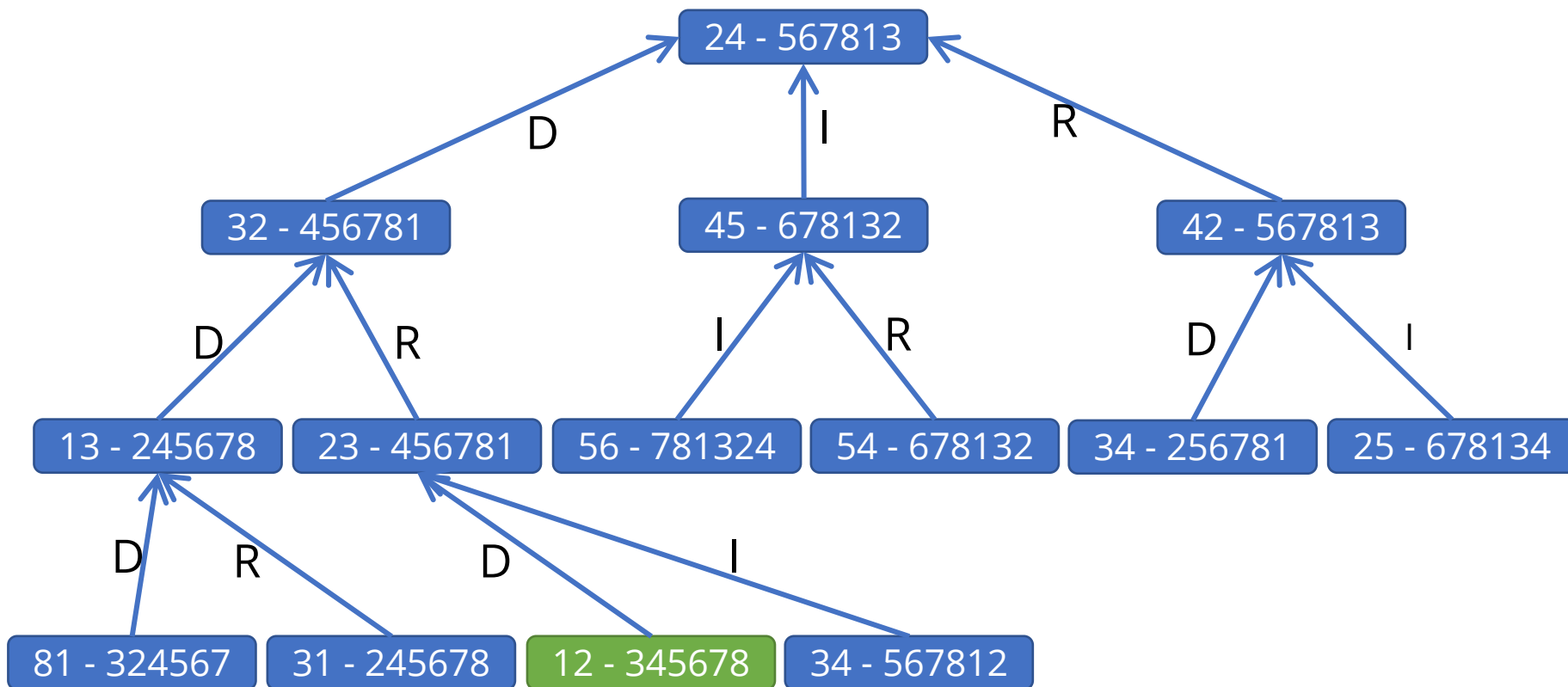
## Solución en amplitud



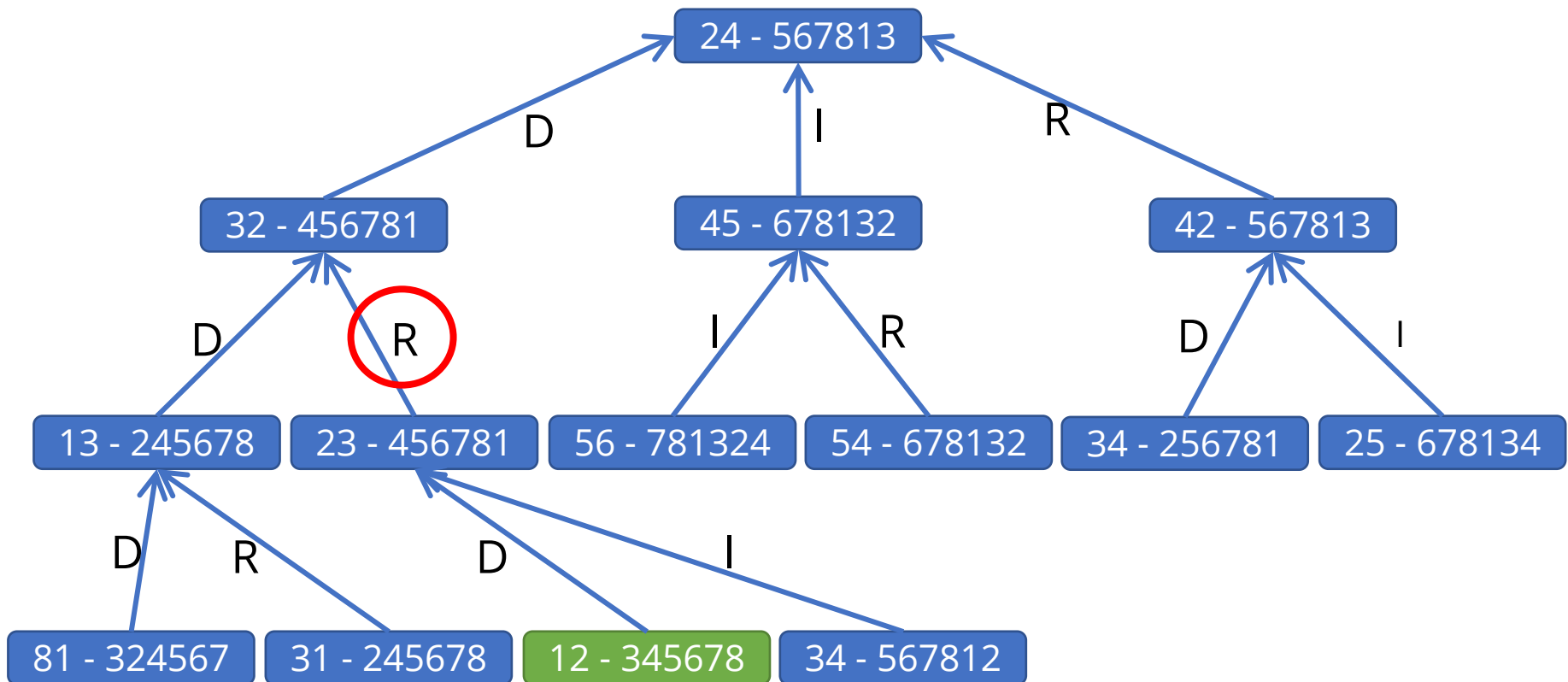
# Ejercicio 5

## Solución en amplitud



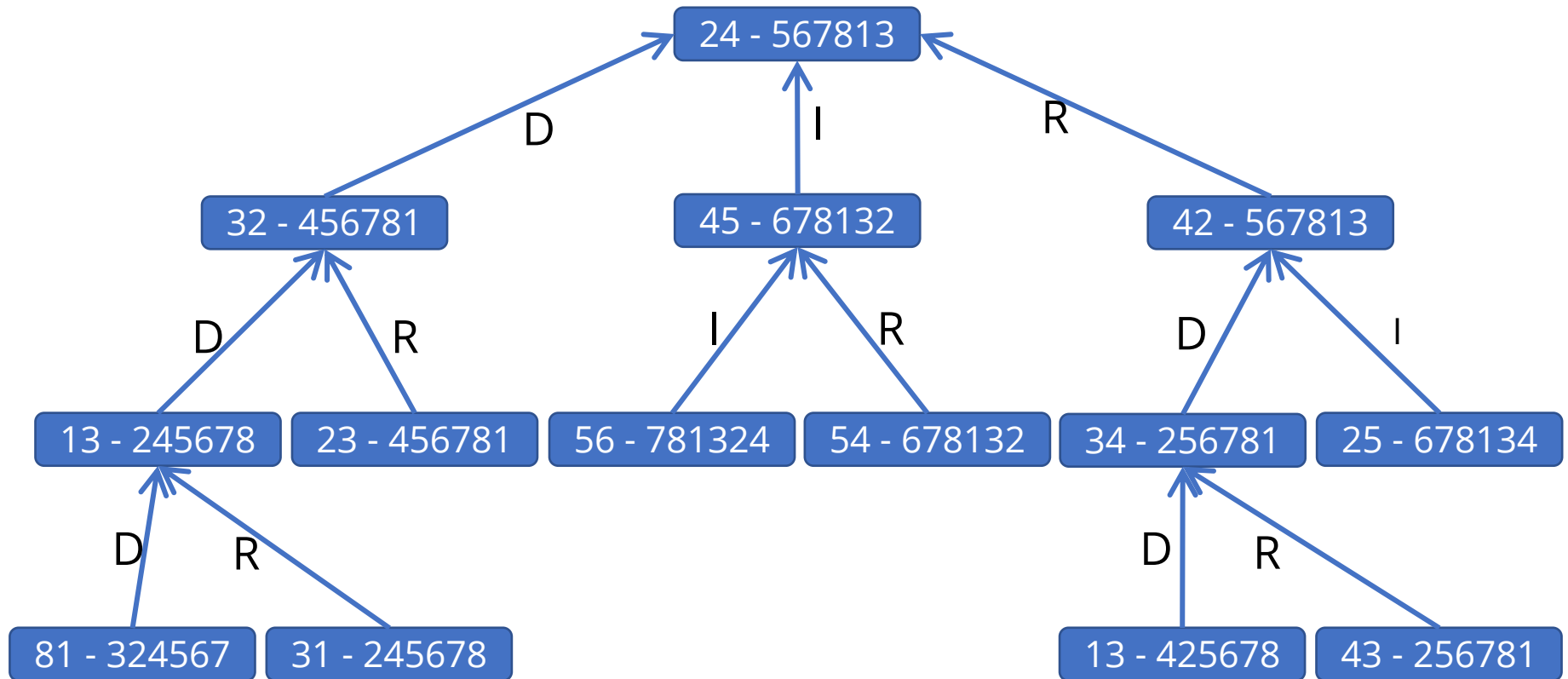


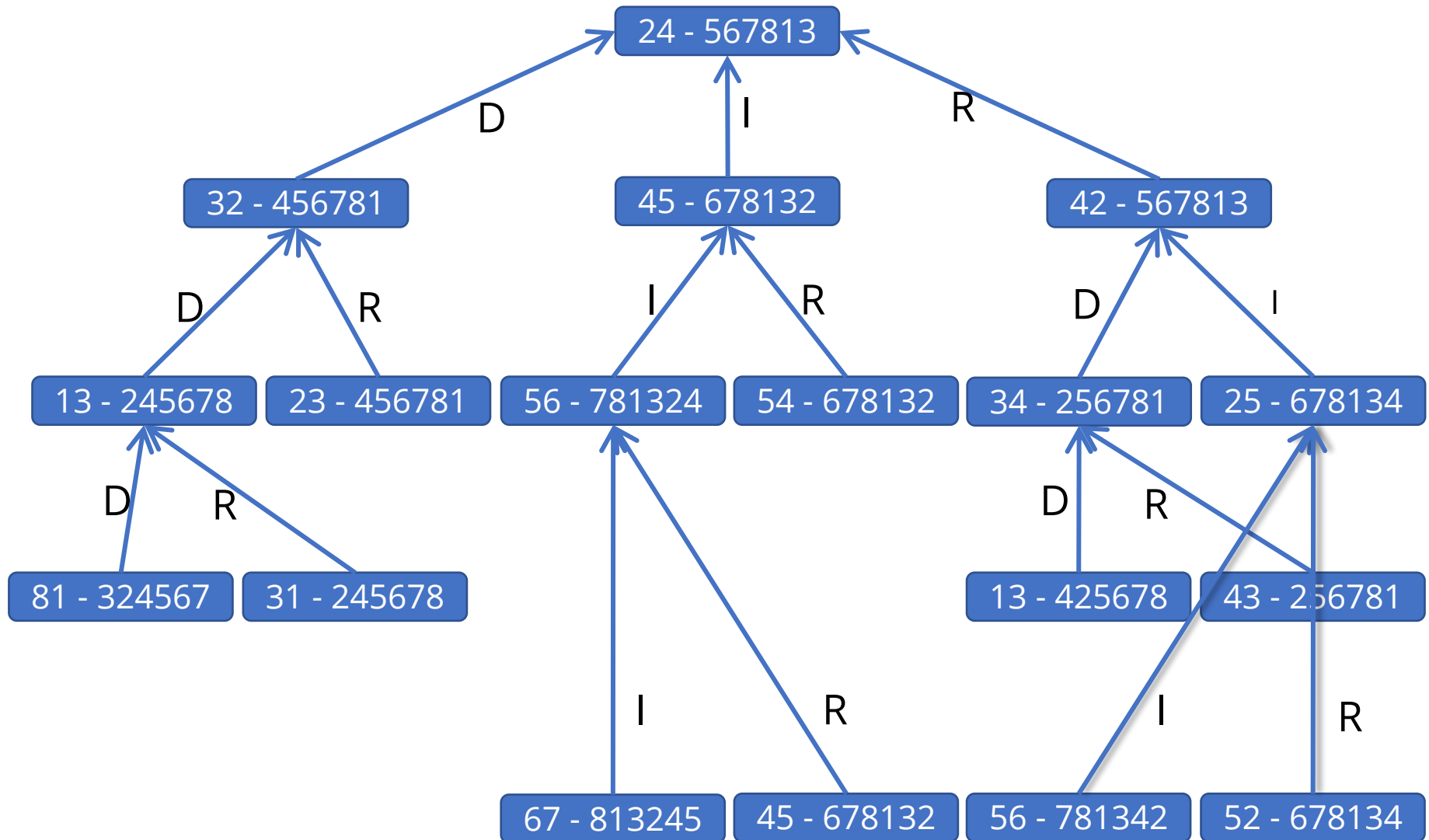


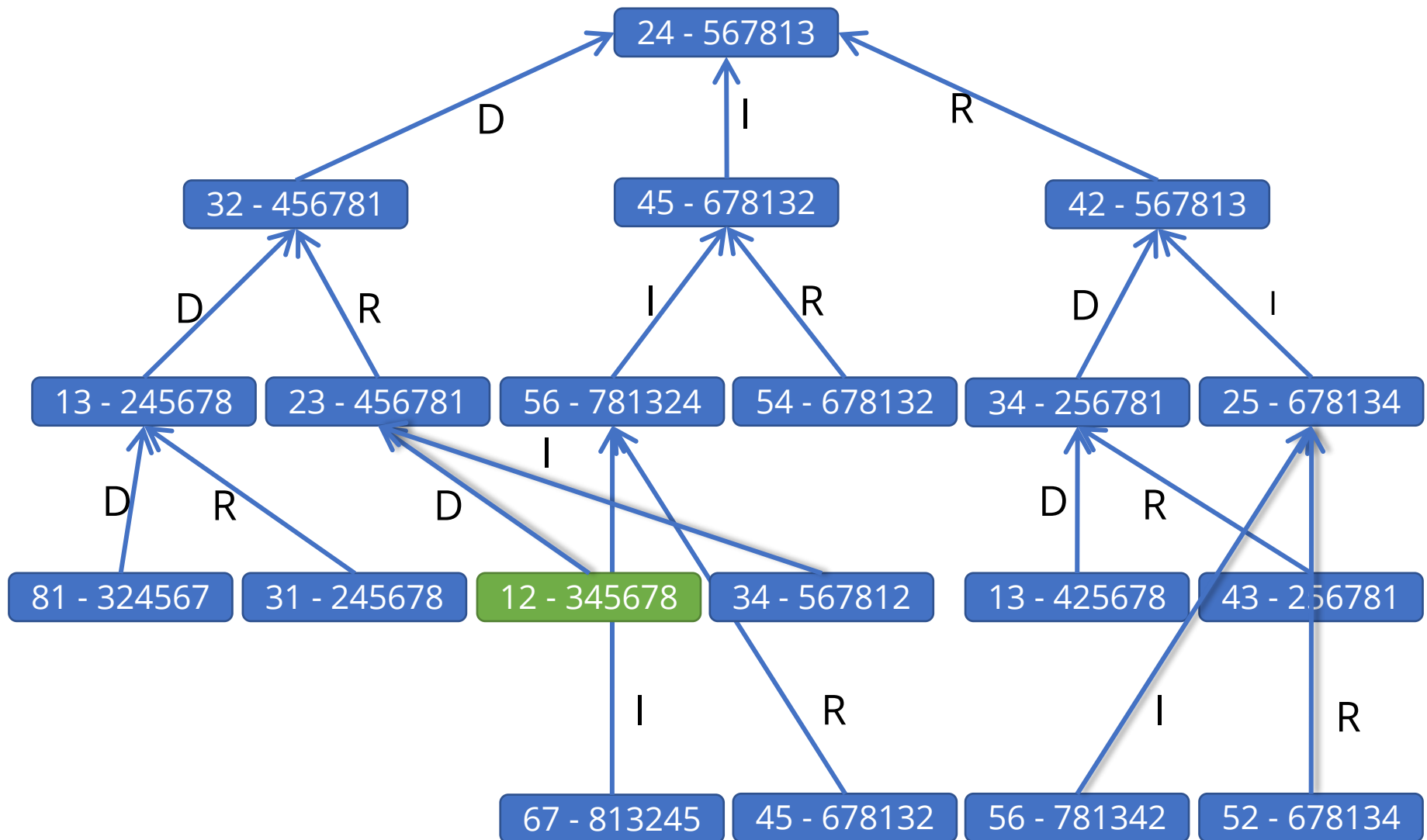


**Error! Recuerda la prioridad: En la misma profundidad:**

1. Mover números a la derecha
2. Mover números a la izquierda
3. Rotar base giratoria

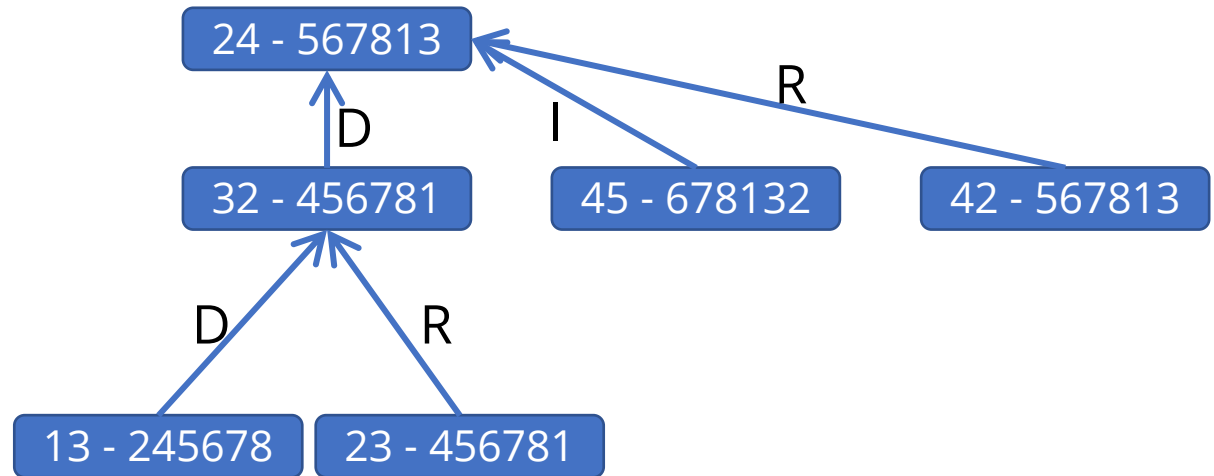


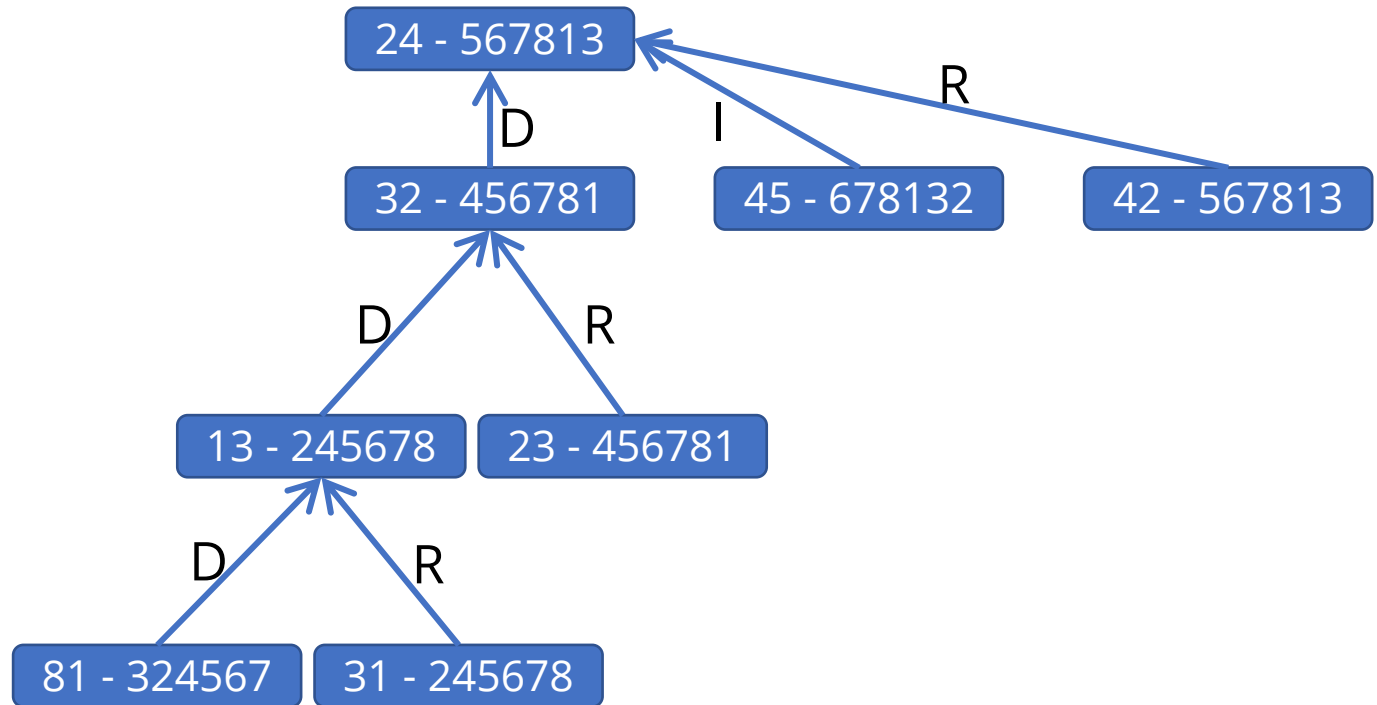




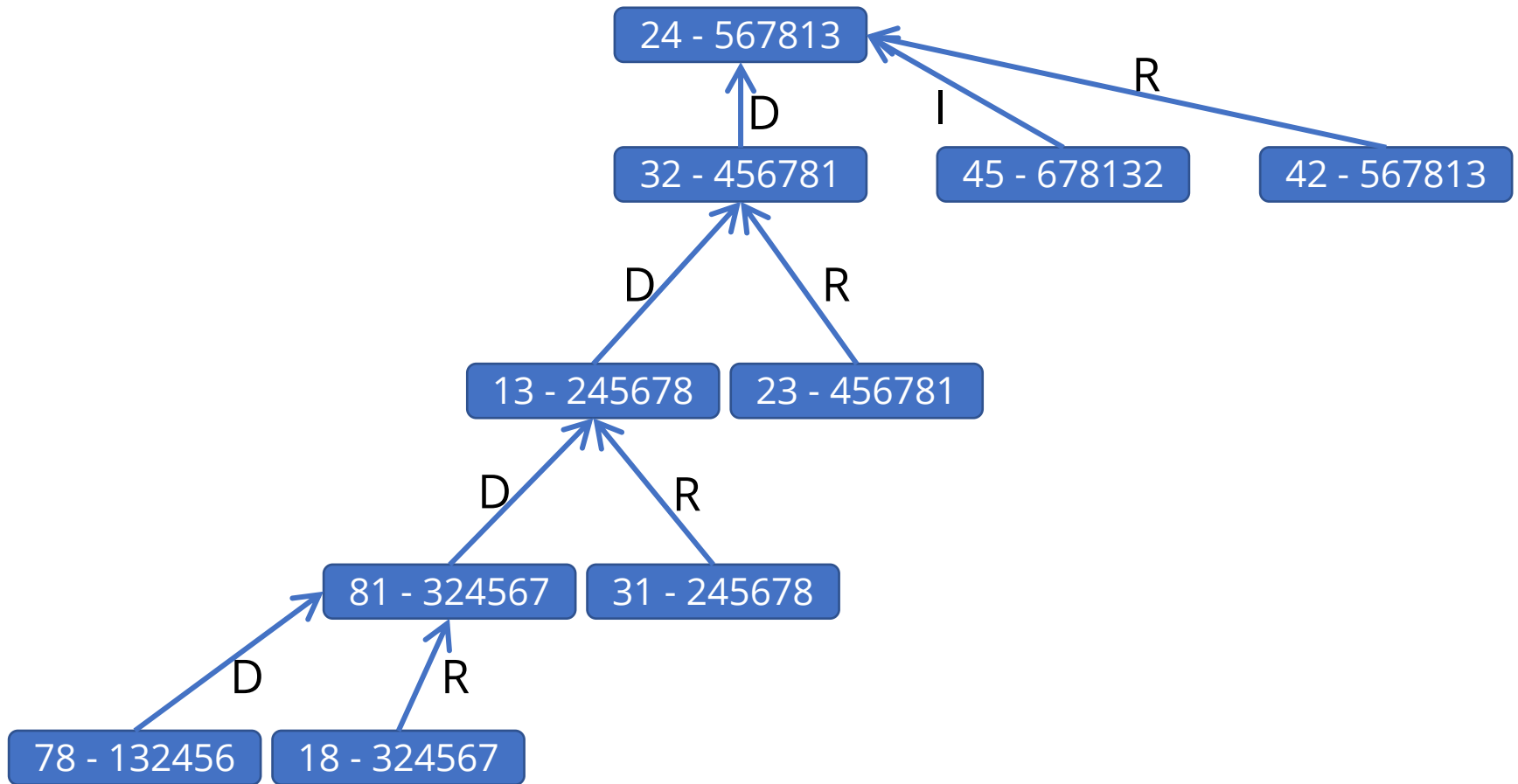
24 - 567813

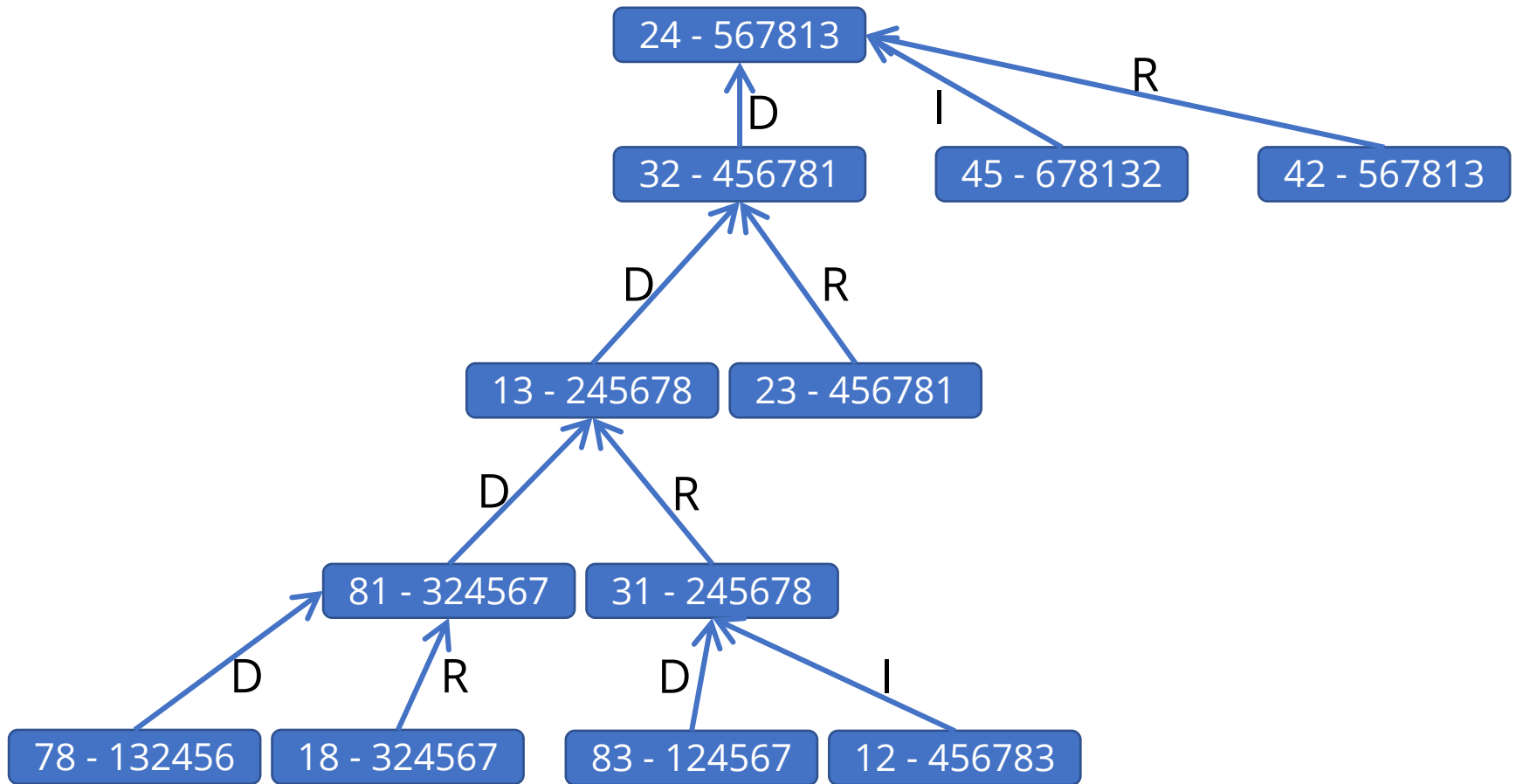


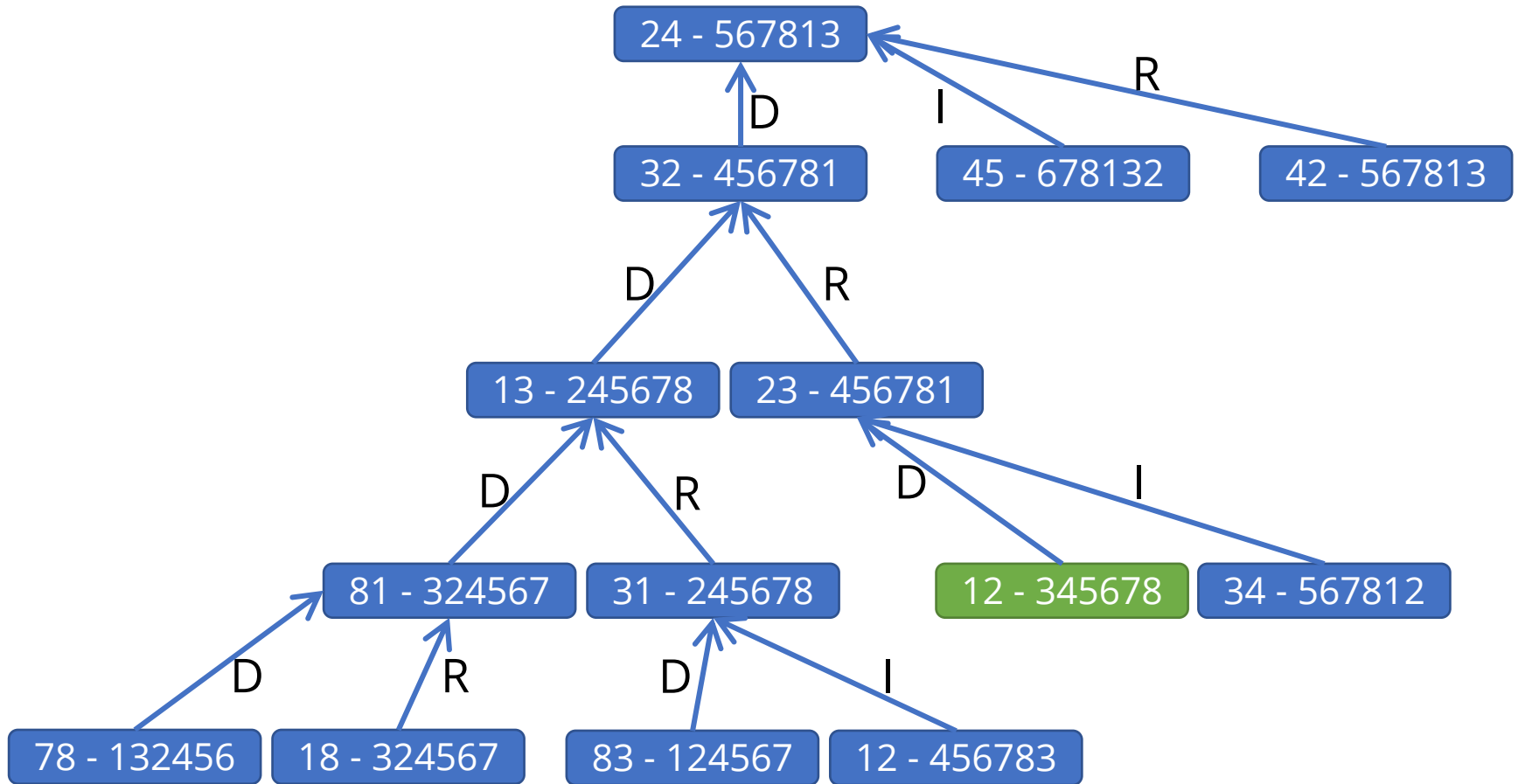












## Ejercicio 6

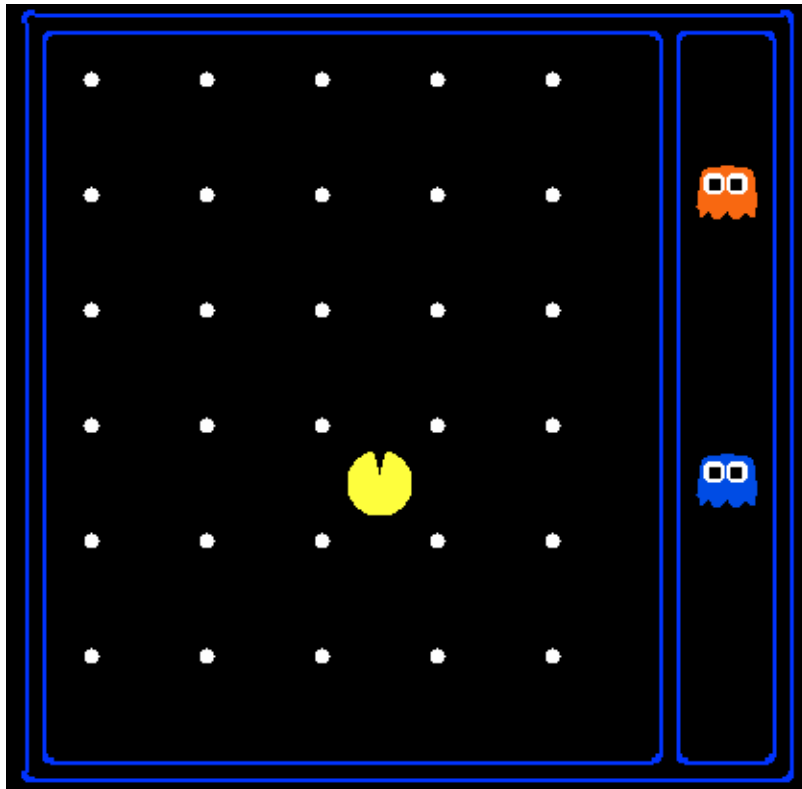
# Enunciado 1

Pág.116

Dada la siguiente imagen de un entorno de juego de Pacman, responda:

### Preguntas:

- ¿Este problema es un problema de búsqueda?
- Defina un estado del problema.
- ¿Cuáles y cómo son las acciones posibles de Pacman?
- ¿Qué tamaño tiene el espacio de estados?



	1	2	3	4	5	6	7	8	9	10		1
1	.		.		.		.		.			
2												
3	.		.		.		.		.			F
4												
5	.		.		.		.		.			
6												
7	.		.		.		.		.			
8						P						F
9	.		.		.		.		.			
10												
11	.		.		.		.		.			
12												
	1	2	3	4	5	6	7	8	9	10		1

**¿Este problema es un problema de búsqueda?**

**Defina un estado del problema.**

**¿Cuáles y cómo son las acciones de Pacman?**

**¿Este problema es un problema de búsqueda?**

No, no tiene ningún objetivo concreto tal y como se ha mostrado la imagen, luego no se podría llegar a ninguna meta.

**Defina un estado del problema.**

**¿Cuáles y cómo son las acciones de Pacman?**

### **¿Este problema es un problema de búsqueda?**

No, no tiene ningún objetivo concreto tal y como se ha mostrado la imagen, luego no se podría llegar a ninguna meta.

### **Defina un estado del problema.**

Representar aquello que caracteriza un estado del mundo:

- Posición del Pacman. X, Y. Valores enteros.
- Dirección hacia la que mira. 4 posibilidades: norte, sur, este, oeste
- Si hay o no comida en cada una de las posiciones donde inicialmente hay comida. Booleano para cada posición.
- posición de cada uno de los fantasmas. Según implementación, XY, booleano...

Estado: (Xpacman, Ypacman, DirPacman, Comida11, Comida13, ..., Comida31, ..., FantNar, FantAzul).

### **¿Cuáles y cómo son las acciones de Pacman?**

### **¿Este problema es un problema de búsqueda?**

No, no tiene ningún objetivo concreto tal y como se ha mostrado la imagen, luego no se podría llegar a ninguna meta.

### **Defina un estado del problema.**

Representar aquello que caracteriza un estado del mundo:

- Posición del Pacman. X, Y. Valores enteros.
- Dirección hacia la que mira. 4 posibilidades: norte, sur, este, oeste
- Si hay o no comida en cada una de las posiciones donde inicialmente hay comida. Booleano para cada posición.
- posición de cada uno de los fantasmas. Según implementación, XY, booleano...

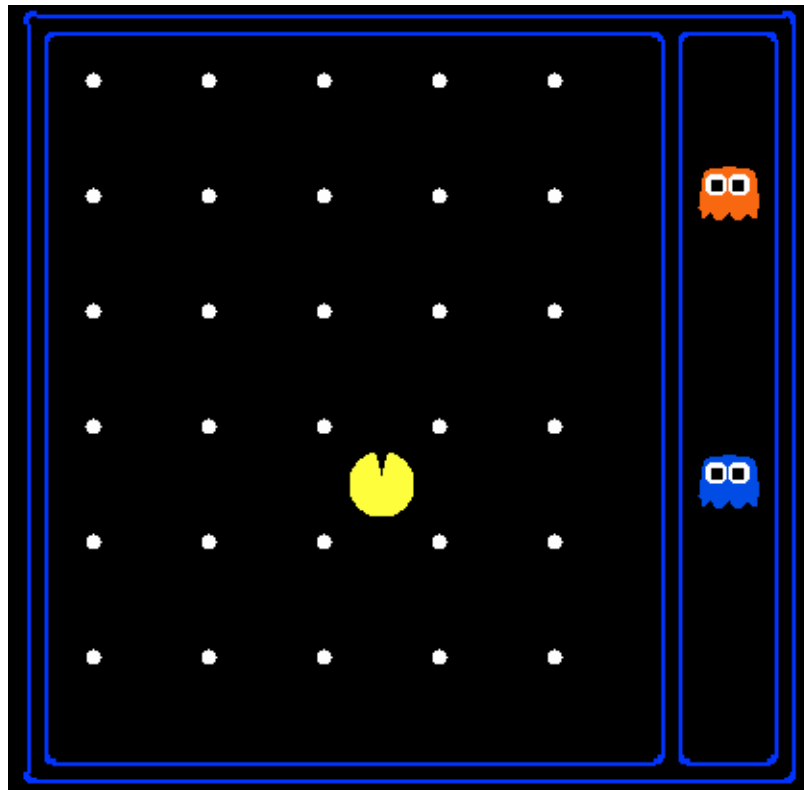
Estado: (Xpacman, Ypacman, DirPacman, Comida11, Comida13, ..., Comida31, ..., FantNar, FantAzul).

### **¿Cuáles y cómo son las acciones de Pacman?**

Acciones: N, S, E, O. Las acciones desplazan al Pacman una posición en la dirección que indica la acción.



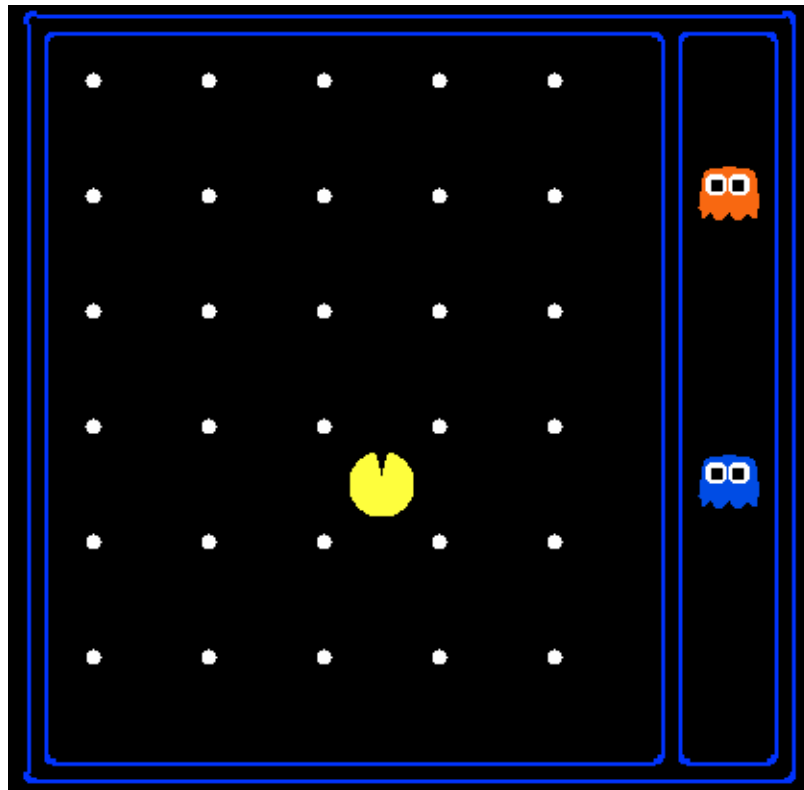
¿Qué tamaño tiene el espacio de estados?



	1	2	3	4	5	6	7	8	9	10		1
1	.		.		.		.		.			
2												
3	.		.		.		.		.			F
4												
5	.		.		.		.		.			
6												
7	.		.		.		.		.			
8						P						F
9	.		.		.		.		.			
10												
11	.		.		.		.		.			
12												
	1	2	3	4	5	6	7	8	9	10		1

### ¿Qué tamaño tiene el espacio de estados?

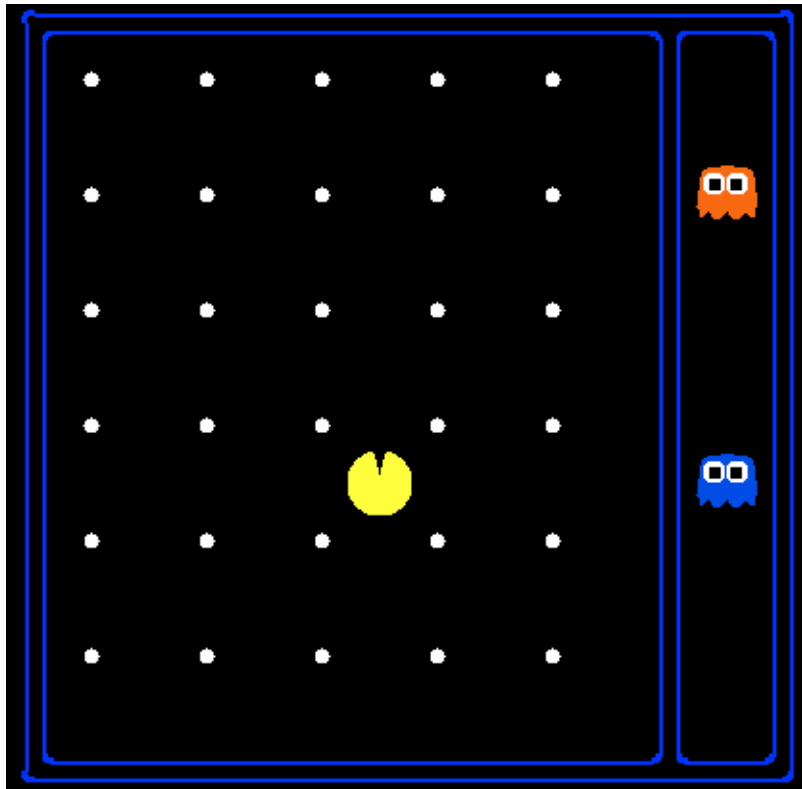
- Posiciones del agente: 10columnas x 12filas = 120
- Posiciones con comidas: 5comidas x 6filas = 30
- Posiciones de cada fantasma: 1columna x 12filas = 12
- Dirección hacia la que mira el Pacman: 4. Norte, sur, este, oeste



	1	2	3	4	5	6	7	8	9	10		1
1	.		.		.		.		.			
2												
3	.		.		.		.		.			F
4												
5	.		.		.		.		.			
6												
7	.		.		.		.		.			
8						P						F
9	.		.		.		.		.			
10												
11	.		.		.		.		.			
12												
	1	2	3	4	5	6	7	8	9	10		1

### ¿Qué tamaño tiene el espacio de estados?

- Posiciones del agente: 10columnas x 12filas = 120
- Posiciones con comidas: 5comidas x 6filas = 30
- Posiciones de cada fantasma: 1columna x 12filas = 12
- Dirección hacia la que mira el Pacman: 4. Norte, sur, este, oeste



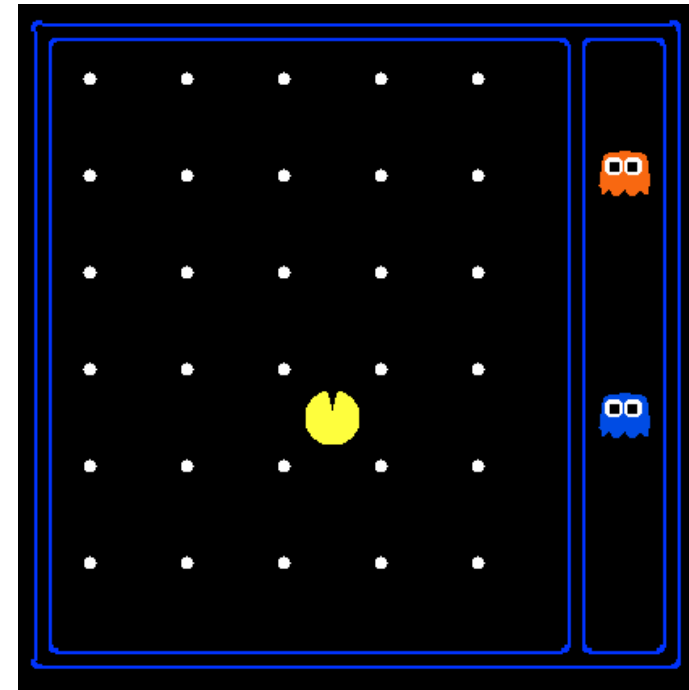
Solución:

Si consideramos todos, hay  
 $120 \times (230) \times (122) \times 4$  estados

Supongamos ahora que en este entorno se define un problema en el que el Pacman comienza en una posición inicial y debe llegar a una meta.

### Preguntas:

- ¿Este problema es un problema de búsqueda?
- Defina un estado del problema.
- ¿Cuáles son el estado inicial y el/los estados finales?
- ¿Cuáles y cómo son las acciones posibles de Pacman?
- ¿Qué tamaño tiene el espacio de estados?



**¿Este problema es un problema de búsqueda?**

**Defina un estado del problema.**

**¿Cuáles son el estado inicial y el/los estados finales?**

**¿Cuáles y cómo son las acciones de Pacman?**

**¿Qué tamaño tiene el espacio de estados?**

**¿Este problema es un problema de búsqueda?**

Sí, puesto que se pueden definir un estado inicial, estado final y acciones para llegar del primero al segundo.

**Defina un estado del problema.**

**¿Cuáles son el estado inicial y el/los estados finales?**

**¿Cuáles y cómo son las acciones de Pacman?**

**¿Qué tamaño tiene el espacio de estados?**

---

**¿Este problema es un problema de búsqueda?**

Sí, puesto que se pueden definir un estado inicial, estado final y acciones para llegar del primero al segundo.

**Defina un estado del problema.**

En este problema únicamente importa la posición de Pacman, por lo que el estado no necesita representar mucha de la información del mundo.

- Posición del Pacman. X, Y. Valores enteros.

Estado: (Xpacman, Ypacman).

**¿Cuáles son el estado inicial y el/los estados finales?**

**¿Cuáles y cómo son las acciones de Pacman?**

**¿Qué tamaño tiene el espacio de estados?**

### **¿Este problema es un problema de búsqueda?**

Sí, puesto que se pueden definir un estado inicial, estado final y acciones para llegar del primero al segundo.

### **Defina un estado del problema.**

En este problema únicamente importa la posición de Pacman, por lo que el estado no necesita representar mucha de la información del mundo.

- Posición del Pacman. X, Y. Valores enteros.

Estado: (Xpacman, Ypacman).

### **¿Cuáles son el estado inicial y el/los estados finales?**

Estado inicial: (Xpacman, Ypacman). Estado final: (Xpacman, Ypacman).

### **¿Cuáles y cómo son las acciones de Pacman?**

### **¿Qué tamaño tiene el espacio de estados?**



### **¿Este problema es un problema de búsqueda?**

Sí, puesto que se pueden definir un estado inicial, estado final y acciones para llegar del primero al segundo.

### **Defina un estado del problema.**

En este problema únicamente importa la posición de Pacman, por lo que el estado no necesita representar mucha de la información del mundo.

- Posición del Pacman. X, Y. Valores enteros.

Estado: (Xpacman, Ypacman).

### **¿Cuáles son el estado inicial y el/los estados finales?**

Estado inicial: (Xpacman, Ypacman). Estado final: (Xpacman, Ypacman).

### **¿Cuáles y cómo son las acciones de Pacman?**

Acciones: N, S, E, O. Las acciones desplazan al Pacman una posición en la dirección que indica la acción.

### **¿Qué tamaño tiene el espacio de estados?**

### **¿Este problema es un problema de búsqueda?**

Sí, puesto que se pueden definir un estado inicial, estado final y acciones para llegar del primero al segundo.

### **Defina un estado del problema.**

En este problema únicamente importa la posición de Pacman, por lo que el estado no necesita representar mucha de la información del mundo.

- Posición del Pacman. X, Y. Valores enteros.

Estado: (Xpacman, Ypacman).

### **¿Cuáles son el estado inicial y el/los estados finales?**

Estado inicial: (Xpacman, Ypacman). Estado final: (Xpacman, Ypacman).

### **¿Cuáles y cómo son las acciones de Pacman?**

Acciones: N, S, E, O. Las acciones desplazan al Pacman una posición en la dirección que indica la acción.

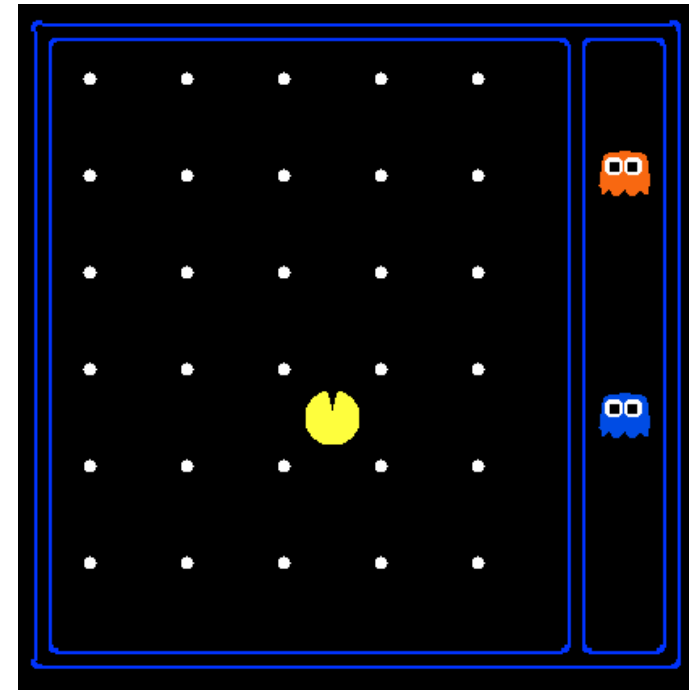
### **¿Qué tamaño tiene el espacio de estados?**

Posiciones del agente: 10columnas x 12filas = 120

Supongamos ahora que en este entorno se define un problema en el que el Pacman debe comer todas las bolas de comida.

**Preguntas:**

- ¿Este problema es un problema de búsqueda?
- Defina un estado del problema.
- ¿Cuáles son el estado inicial y el/los estados finales?
- ¿Cuáles y cómo son las acciones posibles de Pacman?
- ¿Qué tamaño tiene el espacio de estados?



**¿Este problema es un problema de búsqueda?**

**Defina un estado del problema.**

**¿Cuáles son el estado inicial y el/los estados finales?**

**¿Este problema es un problema de búsqueda?**

Sí, puesto que se pueden definir un estado inicial, estado final y acciones para llegar del primero al segundo.

**Defina un estado del problema.**

**¿Cuáles son el estado inicial y el/los estados finales?**

**¿Este problema es un problema de búsqueda?**

Sí, puesto que se pueden definir un estado inicial, estado final y acciones para llegar del primero al segundo.

**Defina un estado del problema.**

En este problema únicamente importa la posición de Pacman, por lo que el estado no necesita representar mucha de la información del mundo.

- Posición del Pacman. X, Y. Valores enteros.
- Si hay o no comida en cada una de las posiciones donde inicialmente hay comida. Booleano para cada posición.

Estado: (Xpacman, Ypacman, Comida11, Comida13, ..., Comida31, ...).

**¿Cuáles son el estado inicial y el/los estados finales?**

**¿Este problema es un problema de búsqueda?**

Sí, puesto que se pueden definir un estado inicial, estado final y acciones para llegar del primero al segundo.

**Defina un estado del problema.**

En este problema únicamente importa la posición de Pacman, por lo que el estado no necesita representar mucha de la información del mundo.

- Posición del Pacman. X, Y. Valores enteros.
- Si hay o no comida en cada una de las posiciones donde inicialmente hay comida. Booleano para cada posición.

Estado: (Xpacman, Ypacman, Comida11, Comida13, ..., Comida31, ...).

**¿Cuáles son el estado inicial y el/los estados finales?**

Estado inicial: (Xpacman, Ypacman, True, True...).

Estado final: (Xpacman, Ypacman, False, False, ...).

**¿Cuáles y cómo son las acciones de Pacman?**

**¿Qué tamaño tiene el espacio de estados?**



**¿Cuáles y cómo son las acciones de Pacman?**

Acciones: N, S, E, O. Las acciones desplazan al Pacman una posición en la dirección que indica la acción.

**¿Qué tamaño tiene el espacio de estados?**

**¿Cuáles y cómo son las acciones de Pacman?**

Acciones: N, S, E, O. Las acciones desplazan al Pacman una posición en la dirección que indica la acción.

**¿Qué tamaño tiene el espacio de estados?**

Posiciones del agente:  $10\text{columnas} \times 12\text{filas} = 120$

Posiciones con comidas:  $5\text{comidas} \times 6\text{filas} = 30$

Entonces hay  $120 \times 2(\text{True o False})^{30}$  estados posibles. Ese es el tamaño del espacio de estados.

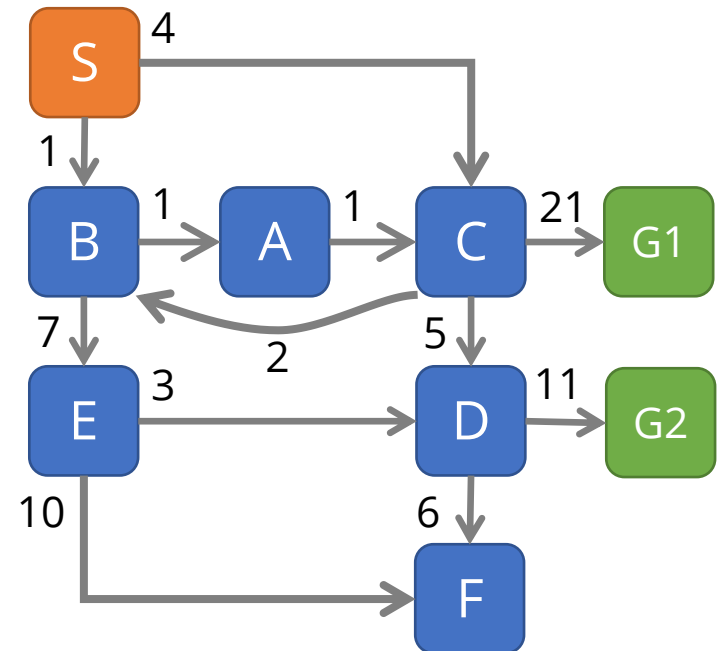
Para este grafo.

- donde los vértices representan estados,
- los arcos son acciones con sus costes asociados,
- el estado inicial es S,
- y el objetivo es llegar a cualquiera de los estados objetivo {G1,G2}.

### Preguntas:

- Hacer Búsqueda en Amplitud.
- Hacer Búsqueda en Profundidad.
- Hacer Búsqueda en Dijkstra.

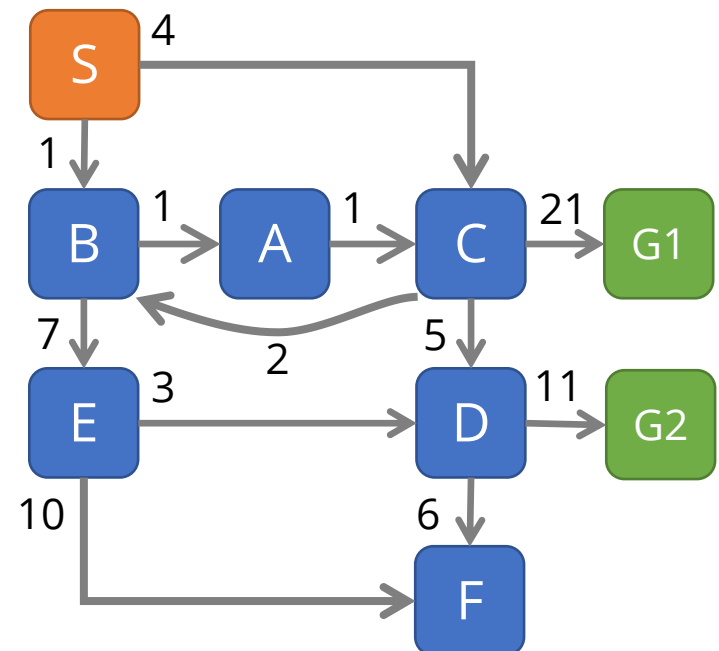
Nota: Para resolver empates, usar el orden alfabético.



Nota: Para resolver empates, usar el orden alfabético.

En amplitud los costes son irrelevantes para el algoritmo, aunque la solución sí tendrá un coste (que lógicamente no garantiza optimalidad).

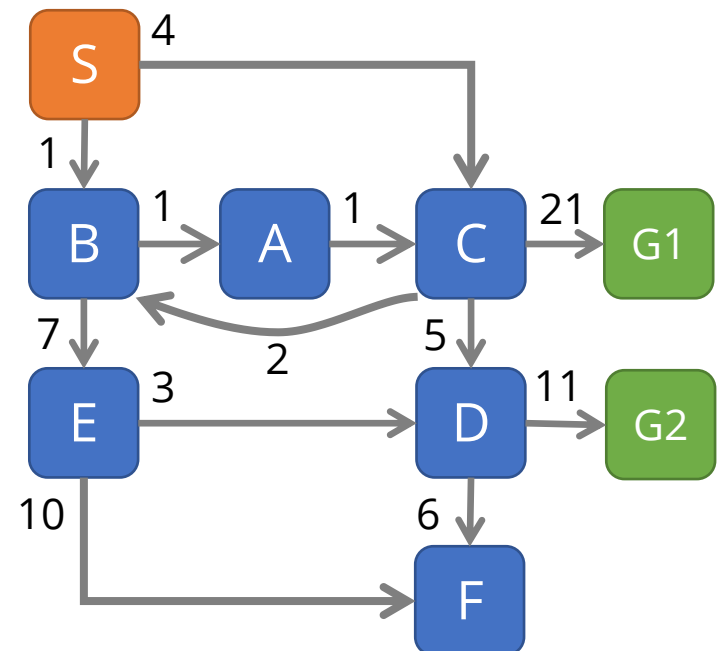
Ciclo	Expande	Genera	Abiertos
0	-	-	S



Nota: Para resolver empates, usar el orden alfabético.

En amplitud los costes son irrelevantes para el algoritmo, aunque la solución sí tendrá un coste (que lógicamente no garantiza optimalidad).

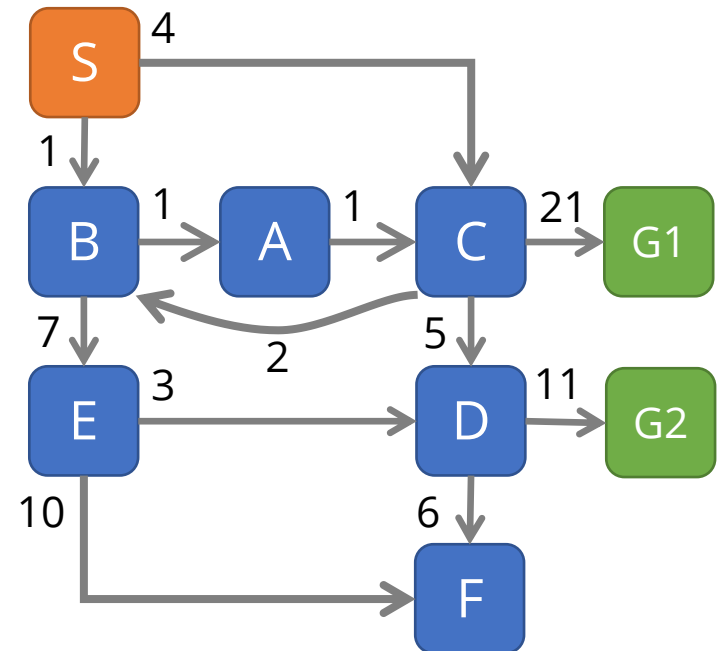
Ciclo	Expande	Genera	Abiertos
0	-	-	S
1	S	B, C	B(1), C(1)



Nota: Para resolver empates, usar el orden alfabético.

En amplitud los costes son irrelevantes para el algoritmo, aunque la solución sí tendrá un coste (que lógicamente no garantiza optimalidad).

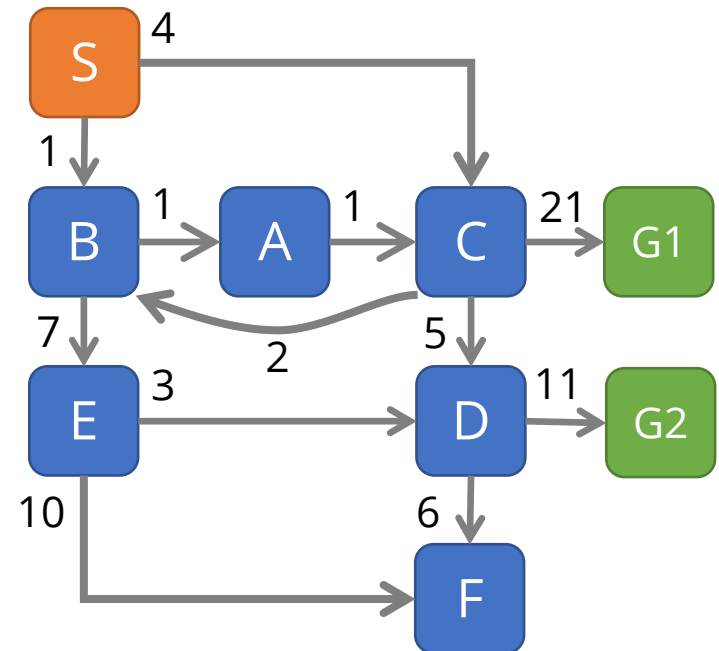
Ciclo	Expande	Genera	Abiertos
0	-	-	S
1	S	B, C	B(1), C(1)
2	B	A, E	C(1), A(2), E(2)



Nota: Para resolver empates, usar el orden alfabético.

En amplitud los costes son irrelevantes para el algoritmo, aunque la solución sí tendrá un coste (que lógicamente no garantiza optimalidad).

Ciclo	Expande	Genera	Abiertos
0	-	-	S
1	S	B, C	B(1), C(1)
2	B	A, E	C(1), A(2), E(2)
3	C	D, B, G1	A, E, D

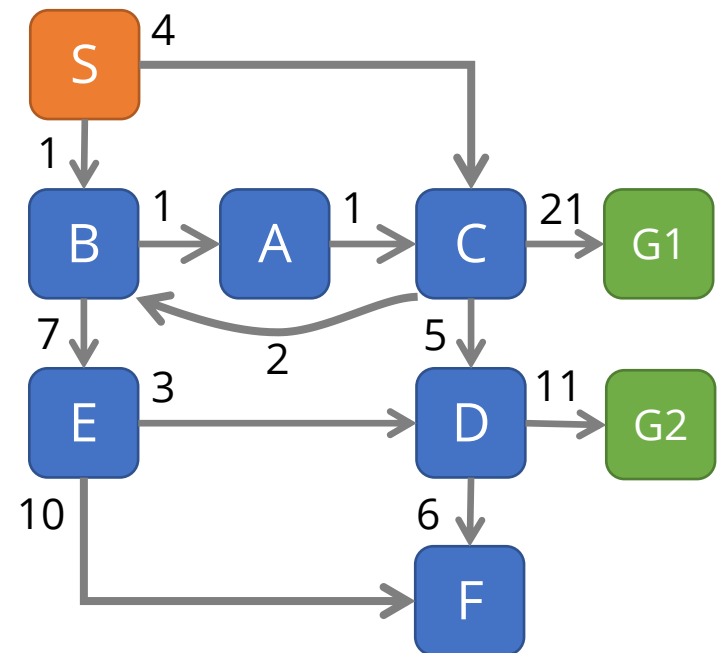


Nota: Para resolver empates, usar el orden alfabético.

En amplitud los costes son irrelevantes para el algoritmo, aunque la solución sí tendrá un coste (que lógicamente no garantiza optimalidad).

Ciclo	Expande	Genera	Abiertos
0	-	-	S
1	S	B, C	B(1), C(1)
2	B	A, E	C(1), A(2), E(2)
3	C	D, <del>B</del> , G1	A(2), E(2), D(3), G1(3)

Nota: el nodo B no se genera al expandir C, porque en amplitud no generamos nodos repetidos (si es repetido un camino más corto le hace llegar).





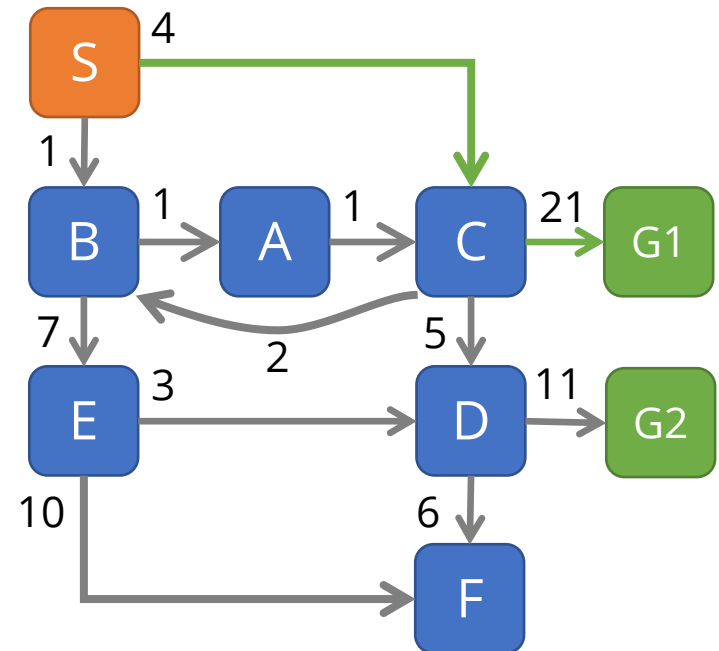
Nota: Para resolver empates, usar el orden alfabético.

En amplitud los costes son irrelevantes para el algoritmo, aunque la solución sí tendrá un coste (que lógicamente no garantiza optimalidad).

Ciclo	Expande	Genera	Abiertos
0	-	-	S
1	S	B, C	B(1), C(1)
2	B	A, E	C(1), A(2), E(2)
3	C	D, B, <b>G1</b>	A(2), E(2), D(3), <b>G1(3)</b>

Camino final: S - C - G1

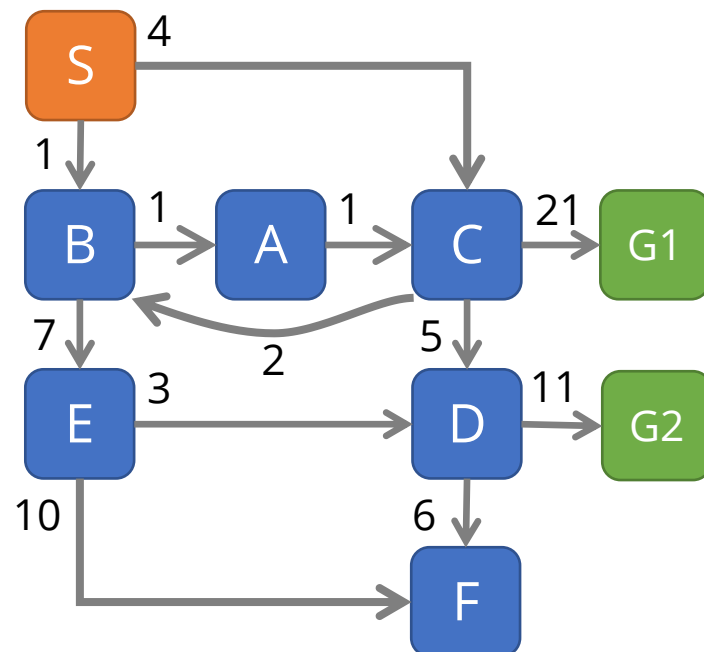
Longitud: 2, Coste: 25



Nota: Para resolver empates, usar el orden alfabético.

En profundidad los costes también son irrelevantes para el algoritmo, aunque la solución sí tendrá un coste (que lógicamente no garantiza optimalidad).

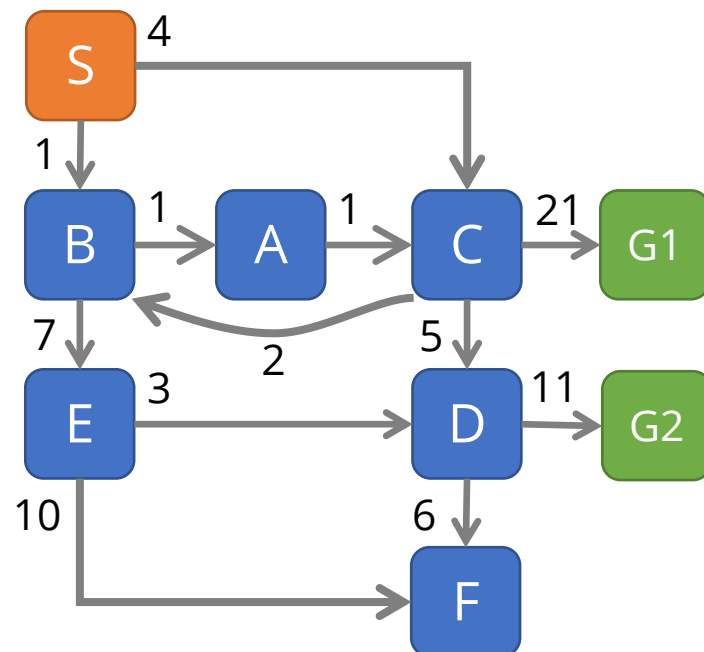
Ciclo	Expande	Genera	Abiertos
0	-	-	S



Nota: Para resolver empates, usar el orden alfabético.

En profundidad los costes también son irrelevantes para el algoritmo, aunque la solución sí tendrá un coste (que lógicamente no garantiza optimalidad).

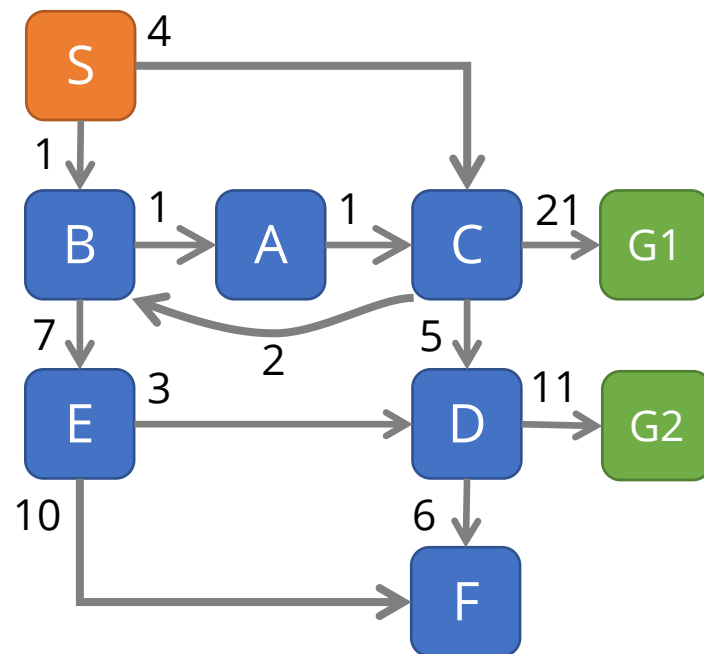
Ciclo	Expande	Genera	Abiertos
0	-	-	S
1	S	B, C	B(1), C(1)



Nota: Para resolver empates, usar el orden alfabético.

En profundidad los costes también son irrelevantes para el algoritmo, aunque la solución sí tendrá un coste (que lógicamente no garantiza optimalidad).

Ciclo	Expande	Genera	Abiertos
0	-	-	S
1	S	B, C	B(1), C(1)
2	B	A, E	C(1), A(2), E(2)

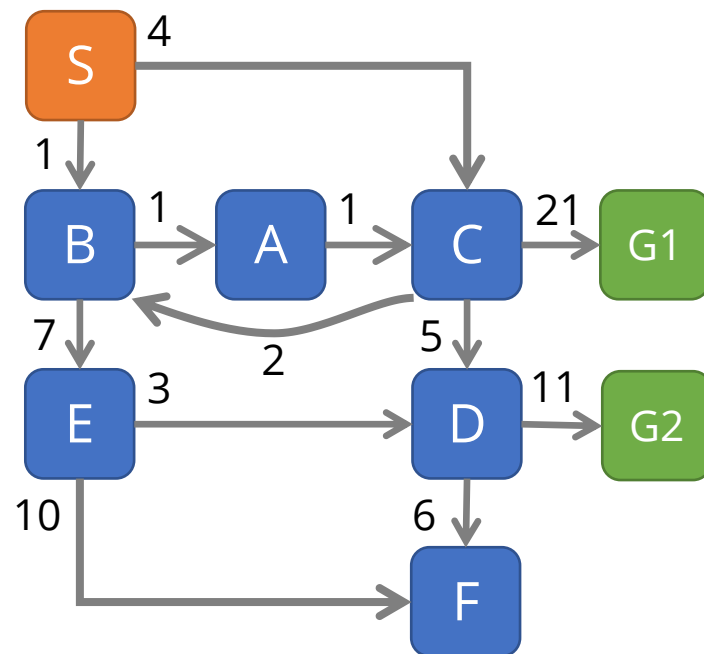


Nota: Para resolver empates, usar el orden alfabético.

En profundidad los costes también son irrelevantes para el algoritmo, aunque la solución sí tendrá un coste (que lógicamente no garantiza optimalidad).

Ciclo	Expande	Genera	Abiertos
0	-	-	S
1	S	B, C	B(1), C(1)
2	B	A, E	C(1), A(2), E(2)
3	A	C	C(1), E(2), C(3)

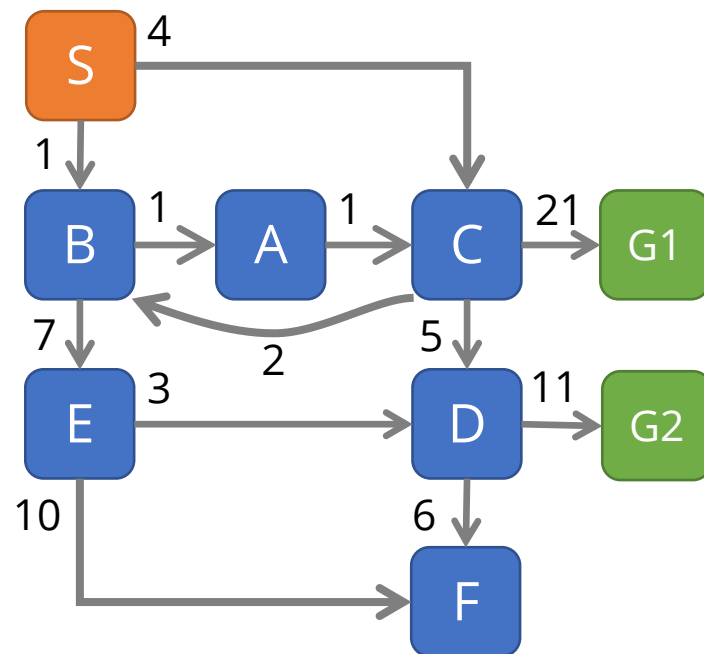
el nodo C(3) se genera al expandir el nodo A, porque asumimos que el algoritmo mete en abierta nodos repetidos cuando no son predecesores del nodo expandido.



Nota: Para resolver empates, usar el orden alfabético.

En profundidad los costes también son irrelevantes para el algoritmo, aunque la solución sí tendrá un coste (que lógicamente no garantiza optimalidad).

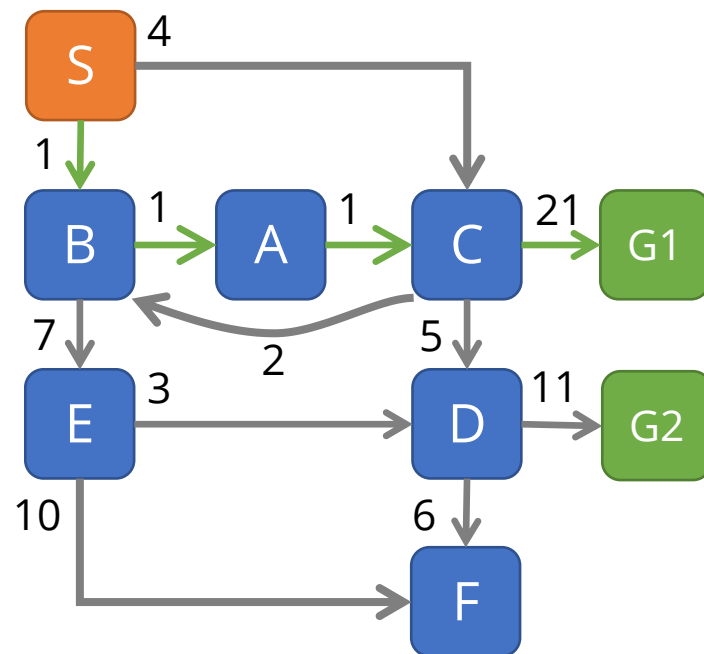
Ciclo	Expande	Genera	Abiertos
0	-	-	S
1	S	B, C	B(1), C(1)
2	B	A, E	C(1), A(2), E(2)
3	A	C	C(1), E(2), C(3)
4	C	B, D, G1	C(1), E(2), B(4), D(4), G1(4)



Nota: Para resolver empates, usar el orden alfabético.

En profundidad los costes también son irrelevantes para el algoritmo, aunque la solución sí tendrá un coste (que lógicamente no garantiza optimalidad).

Ciclo	Expande	Genera	Abiertos
0	-	-	S
1	S	B, C	B(1), C(1)
2	B	A, E	C(1), A(2), E(2)
3	A	C	C(1), E(2), C(3)
4	C	B, D, <b>G1</b>	C(1), E(2), B(4), D(4), <b>G1(4)</b>

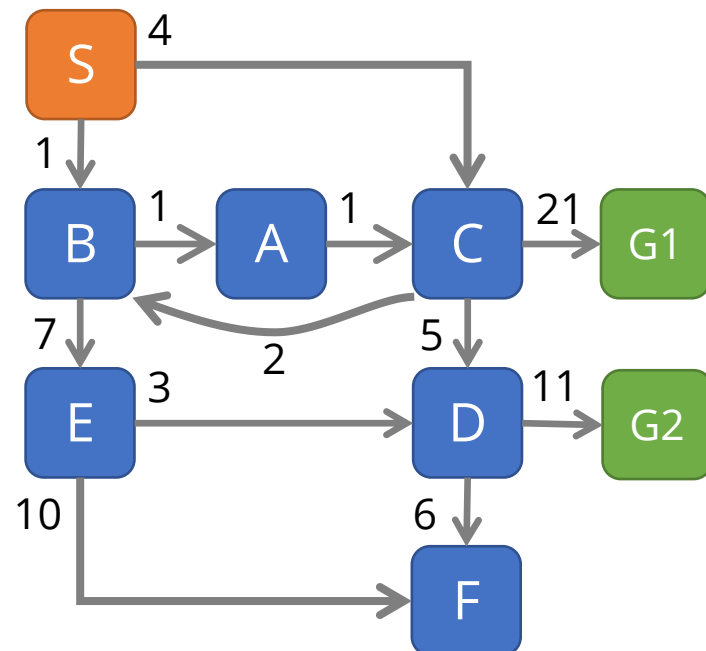


Solución: S - B - A - C - G1, Longitud: 4, Coste: 24

Nota: Para resolver empates, usar el orden alfabético.

En Dijkstra, los costes acumulados se usan para ordenar la lista abierta.

Ciclo	Expande	Genera	Abiertos	Cerrada
0	-	-	S	-

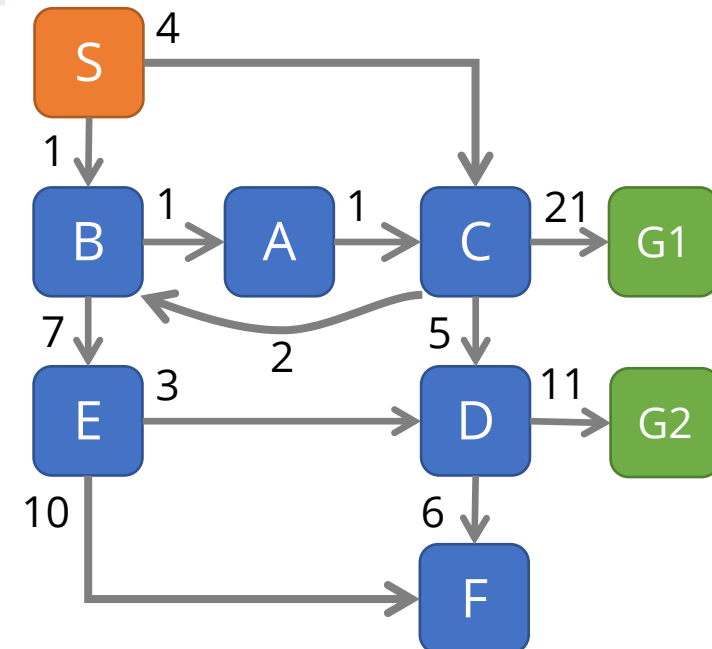




Nota: Para resolver empates, usar el orden alfabético.

En Dijkstra, los costes acumulados se usan para ordenar la lista abierta.

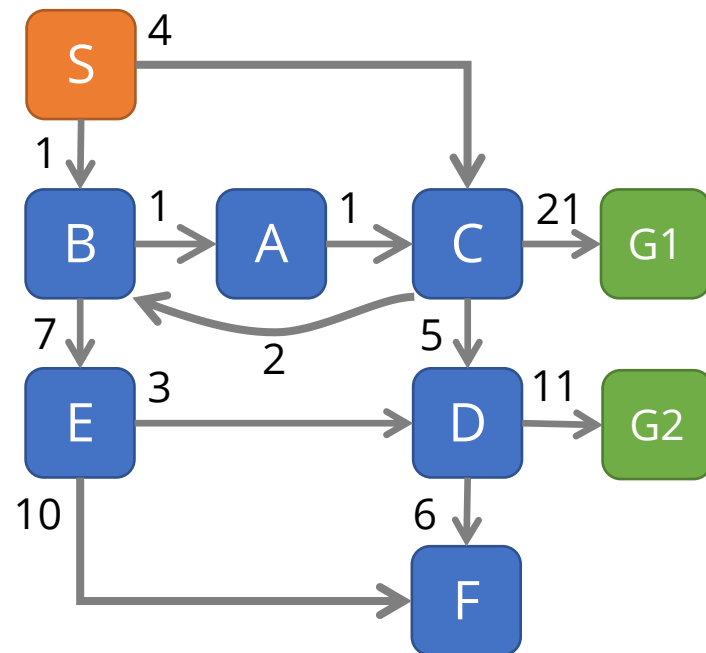
Ciclo	Expande	Genera	Abiertos	Cerrada
0	-	-	S	-
1	S	B, C	B(1), C(4)	S



Nota: Para resolver empates, usar el orden alfabético.

En Dijkstra, los costes acumulados se usan para ordenar la lista abierta.

Ciclo	Expande	Genera	Abiertos	Cerrada
0	-	-	S	-
1	S	B, C	B(1), C(4)	S
2	B	A, E	C(4), A(2), E(8)	S, B(1)

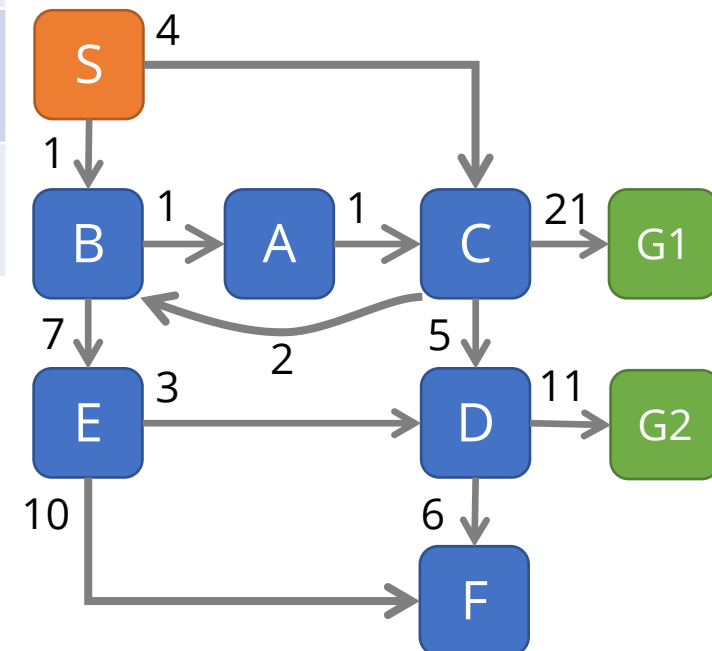


Nota: Para resolver empates, usar el orden alfabético.

En Dijkstra, los costes acumulados se usan para ordenar la lista abierta.

Ciclo	Expande	Genera	Abiertos	Cerrada
0	-	-	S	-
1	S	B, C	B(1), C(4)	S
2	B	A, E	C(4), A(2), E(8)	S, B(1)
3	A	C	C(4), E(8), C(3)	S, B, A

el nodo C(4) se elimina de la lista abierta al aparecer el nodo C(3) con menor coste.

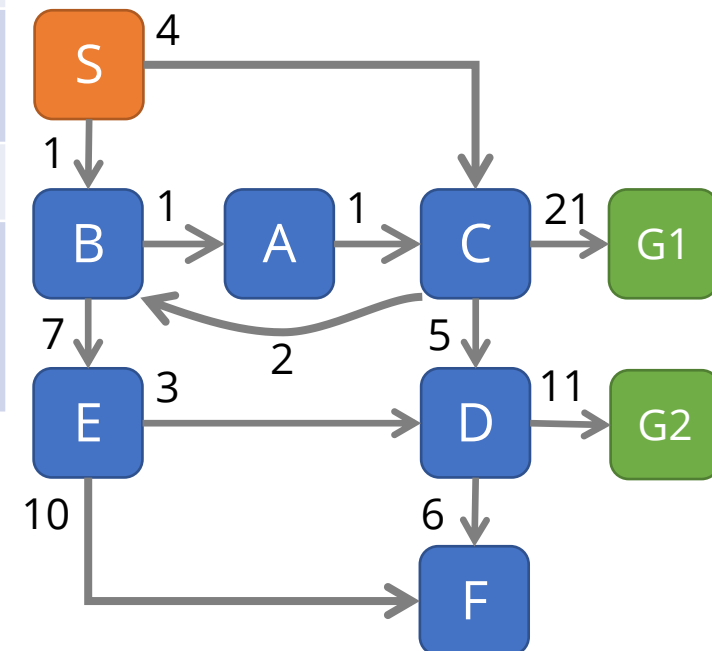


Nota: Para resolver empates, usar el orden alfabético.

En Dijkstra, los costes acumulados se usan para ordenar la lista abierta.

Ciclo	Expande	Genera	Abiertos	Cerrada
0	-	-	S	-
1	S	B, C	B(1), C(4)	S
2	B	A, E	C(4), A(2), E(8)	S, B(1)
3	A	C	E(8), C(3)	S, B, A
4	C	B, D, G1	E(8), B(5), D(8), G1(24)	S, B, A, C

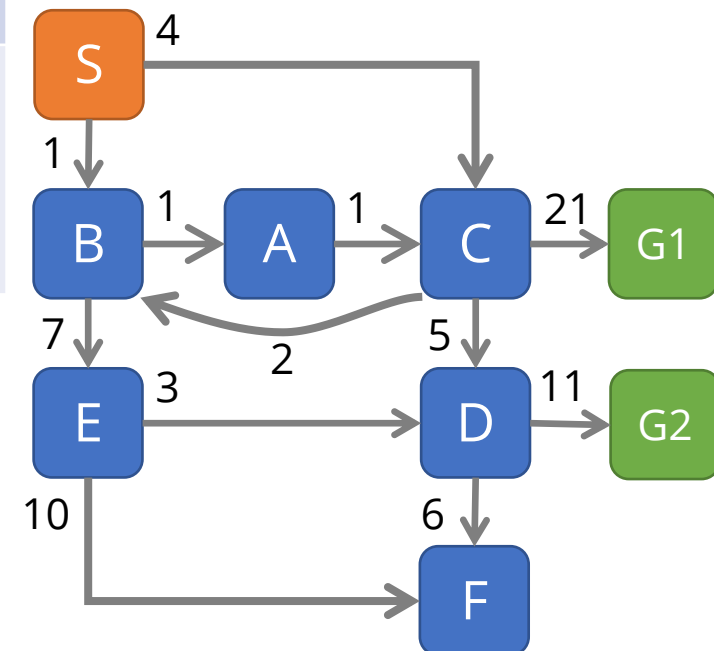
el nodo B(5) no se inserta por estar en la lista CERRADA con menor coste (expandido B(1) en el ciclo 2).



Nota: Para resolver empates, usar el orden alfabético.

En Dijkstra, los costes acumulados se usan para ordenar la lista abierta.

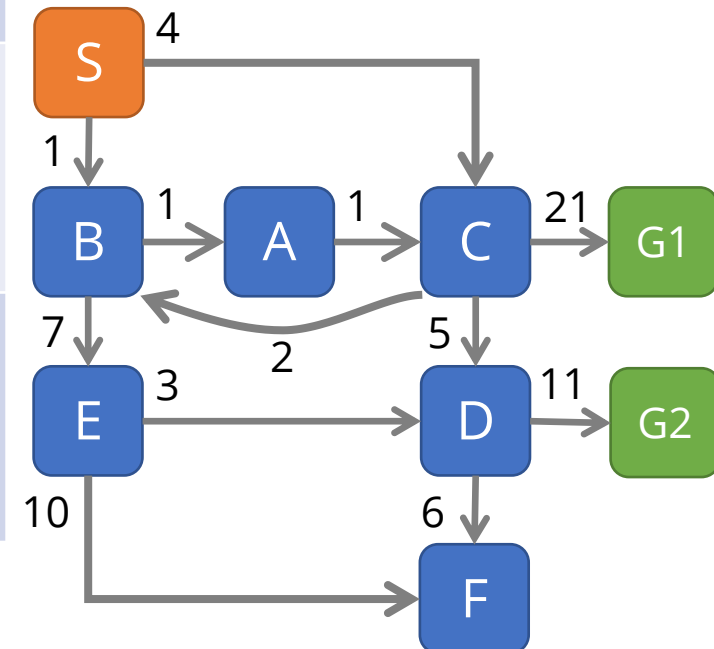
Ciclo	Expande	Genera	Abiertos	Cerrada
4	C	B, D, G1	E(8), B(5), D(8), G1(24)	S, B, A, C
5	D	G2, F	E(8), G1(24), G2(19), F(14)	S, B, A, C, D



Nota: Para resolver empates, usar el orden alfabético.

En Dijkstra, los costes acumulados se usan para ordenar la lista abierta.

Ciclo	Expande	Genera	Abiertos	Cerrada
4	C	B, D, G1	E(8), B(5), D(8), G1(24)	S, B, A, C
5	D	G2, F	E(8), G1(24), G2(19), F(14)	S, B, A, C, D
6	E	D, F	G1(24), G2(19), F(14, 18), D(11)	S, B, A, C, D, E



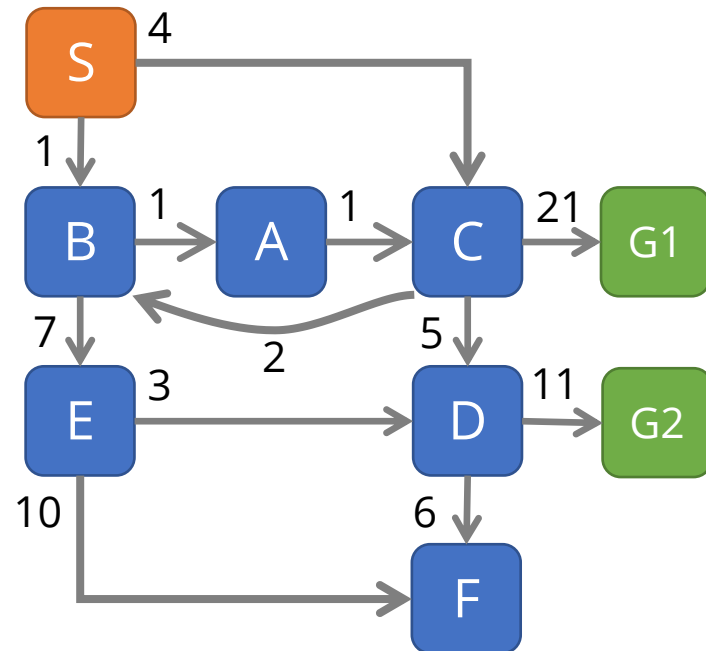
el nodo D(11) no se inserta por estar en la lista CERRADA con menor coste (expandido D(8) en el ciclo 2).

el nodo F(8+10) no se inserta por estar en la lista ABIERTA con menor coste.

Nota: Para resolver empates, usar el orden alfabético.

En Dijkstra, los costes acumulados se usan para ordenar la lista abierta.

Ciclo	Expande	Genera	Abiertos	Cerrada
6	E	D, F	G1(24), G2(19), F(14)	S, B, A, C, D, E
7	F	-	G1(24), G2(19),	S, B, A, C, D, E, F

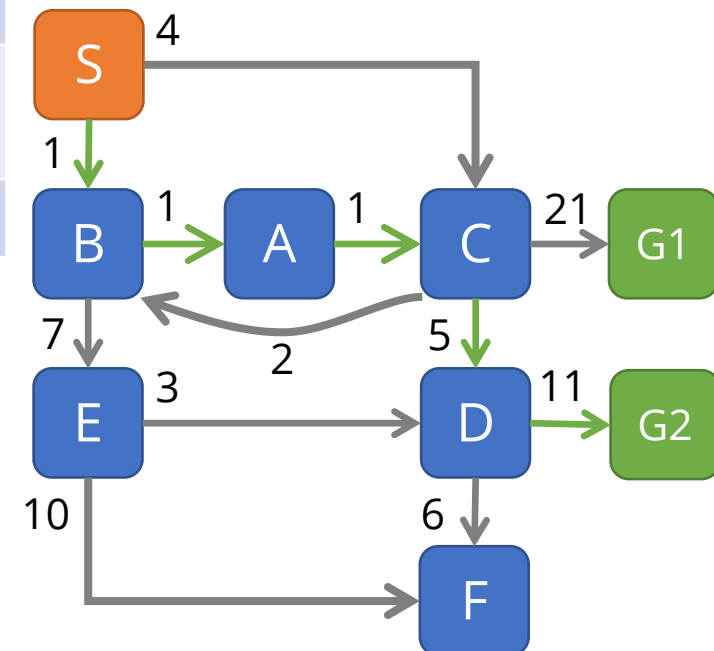


Nota: Para resolver empates, usar el orden alfabético.

En Dijkstra, los costes acumulados se usan para ordenar la lista abierta.

Ciclo	Expande	Genera	Abiertos	Cerrada
6	E	D, F	G1(24), G2(19), F(14)	S, B, A, C, D, E
7	F	-	G1(24), G2(19),	S, B, A, C, D, E, F
8	<b>G2</b>			

Solución: S - B - A - C - D - G2 , Longitud:5, Coste: 19





Suponga que se tiene el siguiente problema: dos robots situados en distintas casillas de una cuadrícula con algunos obstáculos y deben llegar a una misma casilla meta.

Cada robot puede moverse en las cuatro direcciones (norte, sur, este y oeste), o quedarse parado en su casilla. También puede limpiar el obstáculo de cualquier casilla adyacente en esas direcciones.

Ambos robots pueden coincidir en la misma casilla. No está permitido que se sitúen sobre un obstáculo. Ambos robots ejecutan acciones de forma simultánea.

**Resuelva razonadamente las siguientes cuestiones:**

- Cuáles serían el espacio de estados y las acciones para resolver este problema con un algoritmo de búsqueda?
- Codifique el estado y las acciones usando la notación que prefiera de las vistas en clase.
- Escriba un solo ejemplo de acción de movimiento y otro de limpieza.

**Espacio de estados:**

## Espacio de estados:

- Hay tantos estados como contenidos puede haber en cada casilla. Cada casilla puede tener 4 situaciones (0, 1 o 2 robots, o bien un obstáculo).
- Si hay  $M \times N$  casillas, hay  $2^{(M \times N)}$  posibles configuraciones de obstáculos.
- Además, hay que limitar que al menos haya dos huecos sin obstáculos
  - $2^{(M \times N)} - 2$
- Si obviemos los obstáculos, el robot 1 puede estar en  $M \times N$  posiciones distintas, y el robot 2 también en  $M \times N$ .
- Además, limitar que  $r1$  y  $r2$  no aparezcan en la misma celda.
  - $(M \times N) \times (M \times N) - (M \times N)$
- Con estas condiciones (puede haber más), el tamaño del espacio de estados sería:  $((M \times N) \times (M \times N) - (M \times N)) \times 2^{(M \times N)} - 2$

**Espacio de acciones:**

## **Espacio de acciones:**

- Las acciones consisten realmente en pares de acciones, una para cada robot. De ese modo, cada uno puede ejecutar una de las nueve opciones de forma independiente (cinco movimientos o limpiar en cada una de las cuatro direcciones).
- Para la búsqueda cada ciclo se compone de dos acciones, la tomada por el robot 1 y la del robot 2.
  - En total, podría haber un máximo de  $9 \times 9 = 81$  acciones posibles para un estado dado.

¿Es posible en la realidad aplicar las 81 acciones para cada estado?

## **Espacio de acciones:**

- Las acciones consisten realmente en pares de acciones, una para cada robot. De ese modo, cada uno puede ejecutar una de las nueve opciones de forma independiente (cinco movimientos o limpiar en cada una de las cuatro direcciones).
- Para la búsqueda cada ciclo se compone de dos acciones, la tomada por el robot 1 y la del robot 2.
  - En total, podría haber un máximo de  $9 \times 9 = 81$  acciones posibles para un estado dado.

Realmente sólo se puede Mover o Limpiar hacia una casilla en función de si hay o no obstáculo, así que en la práctica el factor de ramificación no puede ser mayor de 25 (cinco para cada robot).

# Codificar el problema

**Codifique el estado:**

**Codifique las acciones:**

## **Codifique el estado:**

Para el estado: podemos usar una tupla por casilla:

(casillaX, casillaY, robot1, robot2, obstáculo)

Donde se distinguen los campos por posición:

- casillaX coordenada X de la casilla casillaY coordenada Y de la casilla.
- robot1 True (o 1) si el robot 1 está en la casilla.
- robot2 True (o 1) si el robot 2 está en la casilla
- obstáculo True (o 1) si la casilla contiene un obstáculo.

## **Codifique las acciones:**



## **Codifique el estado:**

Para el estado: podemos usar una tupla por casilla:

(casillaX, casillaY, robot1, robot2, obstáculo)

Donde se distinguen los campos por posición:

- casillaX coordenada X de la casilla casillaY coordenada Y de la casilla.
- robot1 True (o 1) si el robot 1 está en la casilla.
- robot2 True (o 1) si el robot 2 está en la casilla
- obstáculo True (o 1) si la casilla contiene un obstáculo.

## **Codifique las acciones:**

Para cada robot X podemos definir las acciones como:

- MNX, MSX, MEX, MWX, PARAX: los cinco movimientos del robot.
- LNX, LSX, LEX, LWX: limpiar la casilla correspondiente.

## Ejemplo de turno

Para el estado: podemos usar una tupla por casilla:

(casillaX, casillaY, robot1, robot2, obstáculo)

Para cada robot X podemos definir las acciones como:

- MNX, MSX, MEX, MWX, PARAX: los cinco movimientos del robot.
- LNX, LSX, LEX, LWX: limpiar la casilla correspondiente.

Un ejemplo de turno del problema sería: (LE1,MN2)

Acción robot1 Acción robot2	Explicación acción / condición de la regla
Precondiciones	Explicación de cada precondición
→	
Postcondiciones	Explicación de cada postcondición

# Ejemplo de turno

Para el estado: podemos usar una tupla por casilla:  
(casillaX, casillaY, robot1, robot2, obstáculo)

- Para cada robot X podemos definir las acciones como:
- MNX, MSX, MEX, MWX, PARAX: los cinco movimientos del robot.
  - LNX, LSX, LEX, LWX: limpiar la casilla correspondiente.

Un ejemplo de turno del problema sería: (LE1,MN2)

LE1 MN2		Limpiar Este para el robot 1 Mover Norte para el robot 2
(?casillaX1,?casillaY1,1,?,0)	∧	Robot1 en X1,Y1
(?casillaX1+1,?casillaY1,0,0,1)	∧	Casilla X1+1,Y1 con obstáculo
(?casillaX2,?casillaY2,?R1O,1,0)	∧	Robot2 en X2,Y2
(?casillaX2,?casillaY2+1,?R1D,0,0)	∧	Casilla X2,Y2+1 sin obstáculo
→		
(?casillaX1+1,?casillaY1,0,0,0)	∧	Modificamos destino R1
(?casillaX2,?casillaY2,?R1O,0,0)	∧	Modificamos origen R2
(?casillaX2,?casillaY2-1,?R1D,1,0)		Modificamos destino R2

A partir de este punto se presentan ejercicios sin solución que pueden ser resueltos por el alumno para apoyar el estudio de la asignatura.

Cualquier duda de los mismos o de ejercicios previos puede ser consultada con el profesor de prácticas en el mail: [damigo@inf.uc3m.es](mailto:damigo@inf.uc3m.es) o en la clase de preparación del examen parcial 1.

Se tienen dos jarras de agua: una de cinco litros de capacidad y otra de tres. Ninguna de ellas tiene marcas de medición. Se tiene un grifo que permite llenar las jarras de agua (completas), también se pueden vaciar o traspasar el contenido de una jarra a otra. Si se traspasa, por ejemplo, el contenido máximo de la jarra grande a la pequeña cuando la pequeña no está vacía, la jarra grande acabará con dos litros y la pequeña se llenará. Si la pequeña contuviera ya un litro, la jarra grande acabará con tres litros.

Inicialmente las jarras están vacías y se desea que la jarra de cinco litros de capacidad contenga exactamente cuatro litros.

Formalizar como problema de búsqueda y mostrar cómo ejecutaríamos búsqueda en amplitud para resolverlo.

Ver online en:

<http://www.mathsisfun.com/games/jugs-puzzle.html>

Un grupo de personas tiene que cruzar un puente. Sólo dos personas pueden cruzar a la vez, como mucho, y una debe llevar una antorcha. Sólo disponemos de una antorcha. Además, son de distintas edades y por lo tanto cada una cruza a distinta velocidad: tardan 1, 2, 5 y 10 minutos respectivamente. Inicialmente están todos a un mismo lado del puente.

Formalizar como problema de búsqueda y mostrar cómo ejecutaríamos búsqueda en amplitud para resolverlo.

Ver online en:

<https://www.inwebson.com/demo/cross-the-bridge/>