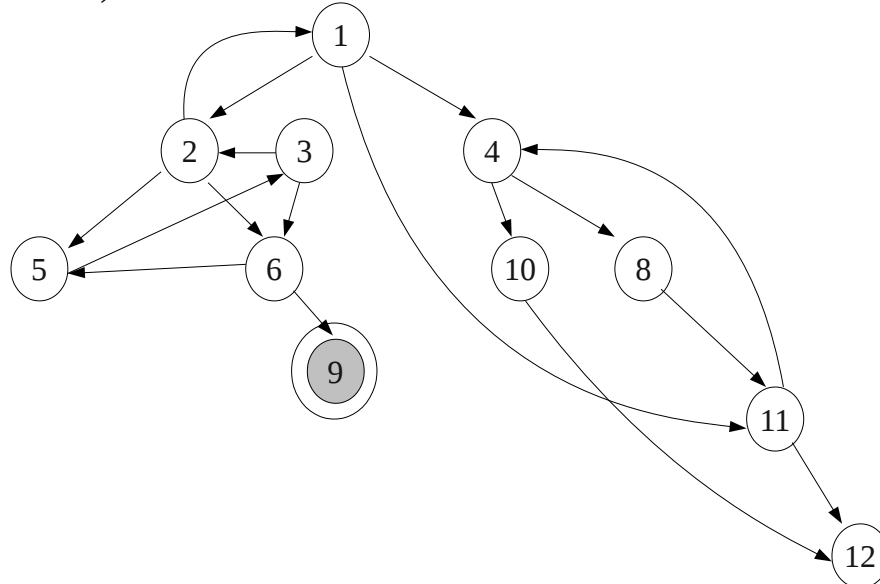


## Relación de problemas acerca de los Métodos de Búsqueda

1. Dado el siguiente espacio de estados, aplica la búsqueda en anchura, en profundidad, bidireccional e iterativa (sólo 10 iteraciones para ésta última). Representa, por cada iteración, el estado de las pilas y del grafo en memoria (aplica el borrado de nodos que utilice cada técnica).



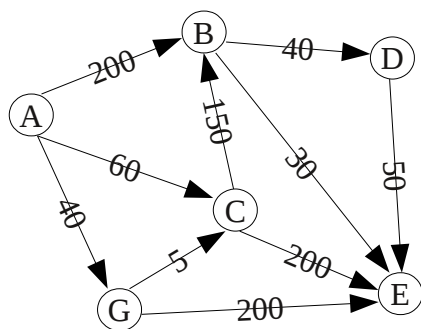
2. Para los siguientes problemas, describa los siguientes aspectos. Cuando el problema concreto se considere simple, resuelva los apartados con un lenguaje formal, en otro caso, describa en lenguaje natural y brevemente la solución a realizar:
  - a) Representación de sus estados.
  - b) Reglas de producción.
  - c) Estrategia de control apropiada para encontrar la solución.

### Problemas

- a) Sudoku
- b) Ajedrez (complejo)
- c) Jarras de agua (simple)
- d) Puzzle a ocho
- e) Las ocho reinas
- f) Recorrido del caballo por el tablero
- g) Problema del viajante sobre un grafo
- h) Torres de Hanoi
- i) Coloreo de grafos (asignar un color a cada nodo de forma que no haya dos nodos vecinos con el mismo color y utilizando sólo 4 colores)
- j) Dentro de un cuarto está un mono y una caja de madera; del techo cuelgan plátanos, fuera del alcance del mono, y éste desea cogerlos.
- k) Igual que en el caso anterior, pero además el mono tiene un palo que tiene que utilizar para poder alcanzar los plátanos
- l) Un granjero quiere cruzar un río con un zorro, un pollo y una bolsa de grano. Él tiene un bote en el que sólo puede llevar a uno de los tres, pero él no puede dejar solos al pollo con el zorro o a la bolsa de grano con el pollo.
- m) **Juego Sol y Sombra:** se dispone de un tablero de 7 casillas y seis fichas, tres de color blanco y tres de color negro, tal como aparece en la figura. El objetivo es intercambiar la colocación de las fichas con el mínimo número de movimientos teniendo en cuenta que:

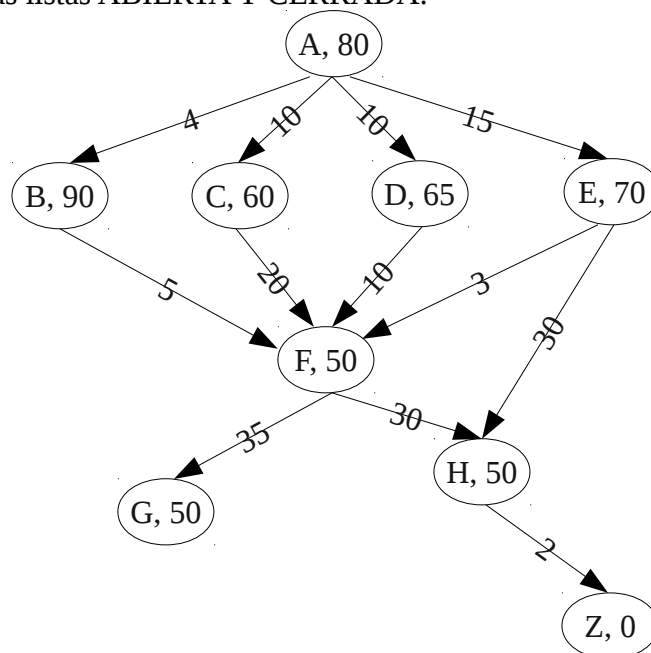
1) No puede haber dos fichas en la misma casilla; 2) Las fichas blancas se desplazan siempre hacia la derecha; las negras sólo hacia la izquierda; 3) Las fichas de uno u otro color pueden realizar uno de los siguientes movimientos: desplazarse a la casilla contigua si está vacía, o saltar por encima de una ficha de otro color siempre que la casilla que haya a continuación esté libre.

3. Resuelva el problema juego Sol y Sombra con los diferentes métodos de búsqueda.
4. Resuelva el problema del puzzle a 8 con los diferentes métodos de búsqueda. Para las búsquedas primero el mejor y A\* utilice como heurística la suma de la distancia de manhattan de cada ficha hasta su posición objetivo.
5. Consideremos un árbol finito de profundidad  $d$  con un factor de ramificación  $b$  (un sólo nodo raíz, con profundidad 0, y  $b$  sucesores por cada nodo, etc.). Supongamos que el nodo objetivo menos profundo se encuentra a una profundidad  $g \leq d$ .
  - a) ¿Cuál es el número mínimo de nodos generados por la búsqueda primero en profundidad con una profundidad límite  $d$ ? ¿Y el máximo?.
  - b) ¿Cuál es el número mínimo de nodos generados por la búsqueda primero en anchura? ¿Y el máximo?.
  - c) ¿Cuál es el número mínimo de nodos generados por el descenso iterativo? ¿Y el máximo? (Considérese que comenzamos con una profundidad límite inicial de 1 y la vamos incrementando una unidad cada iteración).
6. Supongamos que mediante la búsqueda primero en anchura se pueden generar 10000 nodos por segundo y que para almacenar cada nodo son necesarios 100 bytes. Con estos datos ¿cuáles son los requisitos de tiempo y memoria para una búsqueda aplicada a un árbol de búsqueda de profundidad  $d$  y un factor de ramificación de 5? Muestre los resultados del análisis en una tabla.
7. Razona situaciones, y dibuja los espacios de búsqueda asociados (árboles o grafos con nodos iniciales y finales y camino encontrado a la solución), en los que:
  - a) La búsqueda en profundidad encuentre la solución antes que la búsqueda en anchura.
  - b) La búsqueda en anchura encuentre la solución antes que la búsqueda en profundidad.
  - c) La búsqueda en profundidad no encuentre la solución óptima.
  - d) La búsqueda en anchura no encuentre la solución óptima.
  - e) Búsqueda en profundidad sea mejor que Primero el mejor.
  - f) Búsqueda en anchura sea mejor que A\*.
  - g) A\* no encuentre la solución óptima.
  - h) La búsqueda en profundidad y la búsqueda en anchura tengan un funcionamiento equivalente.
  - i) La búsqueda de profundidad iterativa sea peor que la búsqueda en profundidad.
8. Dado el grafo de la figura, determinar el mejor camino para ir del nodo A al E mediante la utilización del algoritmo A\*:



Nodo	Distancia Heurística a E
A	150
B	30
C	70
D	50
G	50
E	0

9. Sea el siguiente grafo, en el que los arcos tienen un coste y los nodos una estimación heurística de su distancia al nodo Z (Z es el nodo objetivo y A es el nodo inicial).
- Sin ningún conocimiento a priori (sin conocer la estructura del grafo, sus pesos...) ¿qué podrías hacer para asegurarte de que A\* encuentra el camino mínimo hasta el nodo solución?
  - Observando el grafo, pero sin aplicar A\* ¿puedes asegurar si este método encontrará o no el camino mínimo entre A y Z?
  - Aplica el algoritmo A\*. Dibuja en cada etapa del algoritmo el subgrafo parcial creado y la situación de las listas ABIERTA Y CERRADA.



10. Responde justificadamente a estas dos cuestiones:
- ¿Por qué la complejidad temporal de todas las búsquedas ciegas, exceptuando la búsqueda bidireccional, es la misma?
  - ¿Por qué la complejidad espacial de la búsqueda en profundidad es menor que la de la búsqueda en anchura?
11. Sea un sistema que para almacenar cada estado de un problema utiliza 1000 bytes. Si se está tratando con un árbol con factor de ramificación de 5, profundidad máxima de 100, y único nodo objetivo, situado en el peor caso, a una profundidad de g (distinta de 100), rellena la siguiente tabla con la memoria máxima que ocupa cada método de búsqueda.
- Nota: Para el caso g, se debe incluir una explicación aparte con figuras que expliquen el razonamiento.

Valor de g	Anchura	Profundidad	Profundidad con retroceso
1			
2			
3			
gg			