

### Procedimiento AMPLITUD/PROFUNDIDAD

1. Crear un árbol de búsqueda A con raíz en s, y una lista de nodos ABIERTOS con s.
2. Crear una lista de nodos CERRADOS vacía.
3. Si ABIERTOS está vacía, entonces devolver 'FRACASO'.
4. Seleccionar  $n \leftarrow \text{primero(ABIERTOS)}$ . Borrar n de ABIERTOS y añadirlo a CERRADOS.
5. Si n es objetivo, entonces devolver el camino de s hasta n en A.
6. Expandir n.  $M \leftarrow \text{sucesores}(n, G) - \text{antecedentes}(n, A)$ .
7. Para cada  $n_2$  en M,
  - a. Si  $n_2$  es nuevo ( $n_2$  no está ABIERTO ni CERRADO),
    - i. Poner un puntero de  $n_2 \rightarrow n$ .
    - ii. Añadir  $n_2$  a ABIERTOS.
  - b. Si  $n_2$  no es nuevo, **ignorarlo**.
8. Ordenar ABIERTOS **por orden de antigüedad** (primero los más antiguos/recientes).
9. Volver a 3.

### Procedimiento COSTE UNIFORME (Dijkstra)

1. Crear un árbol de búsqueda A con raíz en s, y una lista de nodos ABIERTOS con s.
2. Crear una lista de nodos CERRADOS vacía.
3. Si ABIERTOS está vacía, entonces devolver 'FRACASO'.
4. Seleccionar  $n \leftarrow \text{primero(ABIERTOS)}$ . Borrar n de ABIERTOS y añadirlo a CERRADOS.
5. Si n es objetivo, entonces devolver el camino de s hasta n en A.
6. Expandir n.  $M \leftarrow \text{sucesores}(n, G) - \text{antecedentes}(n, A)$ .
7. Para cada  $n_2$  en M,
  - a. Si  $n_2$  es nuevo ( $n_2$  no está ABIERTO ni CERRADO),
    - i. Poner un puntero de  $n_2 \rightarrow n$ .
    - ii. Añadir  $n_2$  a ABIERTOS.
  - b. Si  $n_2$  no es nuevo, **y el valor de  $g(n_2)$  es menor a través del nuevo camino, entonces, redirigir su puntero hacia n.**
8. Ordenar ABIERTOS por **orden creciente en el valor de  $g(n)$** .
9. Volver a 3.

### Procedimiento BÚSQUEDA ÁVIDA

1. Crear un árbol de búsqueda A con raíz en s, y una lista de nodos ABIERTOS con s.
2. Crear una lista de nodos CERRADOS vacía.
3. Si ABIERTOS está vacía, entonces devolver 'FRACASO'.
4. Seleccionar  $n \leftarrow \text{primero(ABIERTOS)}$ . Borrar n de ABIERTOS y añadirlo a CERRADOS.
5. Si n es objetivo, entonces devolver el camino de s hasta n en A.
6. Expandir n.  $M \leftarrow \text{sucesores}(n, G) - \text{antecedentes}(n, A)$ .
7. Para cada  $n_2$  en M,
  - a. Si  $n_2$  es nuevo ( $n_2$  no está ABIERTO ni CERRADO),
    - i. Poner un puntero de  $n_2 \rightarrow n$ .
    - ii. Añadir  $n_2$  a ABIERTOS.
  - b. Si  $n_2$  no es nuevo, **ignorarlo**.
8. Ordenar ABIERTOS **por orden creciente de  $f(n) = h(n)$** .
9. Volver a 3.

### Procedimiento A\*

1. Crear un árbol de búsqueda A con raíz en s, y una lista de nodos ABIERTOS con s.
2. Crear una lista de nodos CERRADOS vacía.
3. Si ABIERTOS está vacía, entonces devolver 'FRACASO'.
4. Seleccionar  $n \leftarrow \text{primero(ABIERTOS)}$ . Borrar n de ABIERTOS y añadirlo a CERRADOS.
5. Si n es objetivo, entonces devolver el camino de s hasta n en A.
6. Expandir n.  $M \leftarrow \text{sucesores}(n, G) - \text{antecedentes}(n, A)$ .
7. Para cada  $n_2$  en M,
  - a. Si  $n_2$  es nuevo ( $n_2$  no está ABIERTO ni CERRADO),
    - i. Poner un puntero de  $n_2 \rightarrow n$ .
    - ii. Añadir  $n_2$  a ABIERTOS.
  - b. Si  $n_2$  no es nuevo, **y el valor de  $g(n_2)$  es menor a través del nuevo camino,**
    - i. **Redirigir su puntero hacia n.**
    - ii. **Si  $n_2$  está en CERRADOS, entonces pasarlo a ABIERTOS.**
8. Ordenar ABIERTOS por **orden creciente en el valor de  $f(n) = g(n) + h(n)$** .
9. Volver a 3.

### Procedimiento BÚSQUEDA-CON-ÁRBOL

1. Crear un árbol de búsqueda A con raíz en s, y una lista de nodos ABIERTOS con s.
2. Crear una lista de nodos CERRADOS vacía.
3. Si ABIERTOS está vacía, entonces devolver 'FRACASO'.
4. Seleccionar  $n \leftarrow \text{primero}(\text{ABIERTOS})$ . Borrar n de ABIERTOS y añadirlo a CERRADOS.
5. Si n es objetivo, entonces devolver el camino de s hasta n en A.
6. Expandir n.  $M \leftarrow \text{sucesores}(n, G) - \text{antecedentes}(n, A)$ .
7. Para cada  $n_2$  en M,
  - a. Si  $n_2$  es nuevo ( $n_2$  no está ABIERTO ni CERRADO),
    - i. Poner un puntero de  $n_2 \rightarrow n$ .
    - ii. Añadir  $n_2$  a ABIERTOS.
  - b. Si  $n_2$  no es nuevo, **“decidir qué camino se conserva en el árbol”**.
8. **“Reordenar”** ABIERTOS.
9. Volver a 3.