



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Cuaderno de trabajo: Búsqueda en profundidad (en árbol)¹

Albert Sanchis

DSIC

Departamento de Sistemas
Informáticos y Computación

¹Para una correcta visualización, se requiere Acrobat Reader v. 7.0 o superior

Objetivos formativos

- ▶ Caracterizar la búsqueda convencional en un grafo de estados.
- ▶ Describir búsqueda en profundidad (en árbol).
- ▶ Construir el árbol de búsqueda en profundidad (en árbol).
- ▶ Aplicar búsqueda en profundidad (en árbol) a un problema clásico.
- ▶ Analizar la calidad de búsqueda en profundidad (en árbol).

Problema: La ruta más corta entre dos puntos

Búsqueda de una ruta más corta desde Arad a Bucarest [1]:



$\text{Acciones}(\text{Arad}) = \{\text{Ir}(\text{Sibiu}), \text{Ir}(\text{Timisoara}), \text{Ir}(\text{Zerind})\}.$

Búsqueda en profundidad [1, 2]

```
DFS( $G, s', m$ )      // Depth-first search con profundidad máxima  $m$   
   $O = \text{IniPila}(s')$       // Open: frontera-pila de la búsqueda  
  mientras no  $\text{PilaVacía}(O)$ :  
     $s = \text{Desapila}(O)$       // selección LIFO (Last in, first out)  
    si  $\text{Objetivo}(s)$  retorna  $s$       // solución encontrada!  
    si  $\text{Profundidad}(s) < m$ :      // no a profundidad máxima  
      para toda  $(s, n) \in \text{Adyacentes}(G, s)$ : // generación:  $n$  hijo de  $s$   
         $\text{Apila}(O, n)$       // añadimos  $n$  a la pila  
  retorna NULL      // ninguna solución encontrada
```

- **Cuestión 1:** Haz una traza del algoritmo **DFS** (en árbol) aplicado al problema de búsqueda de una ruta más corta desde Arad a Bucarest aplicando una profundidad máxima $m = 3$.

O	S
{Arad (p=0)}	—
{Sibiu (p=1), Timisoara (p=1), Zerind (p=1)}	Arad (p=0)
{Arad (p=2), Fagaras (p=2), Oradea (p=2), Rimnicu(p=2), Timisoara (p=1), Zerind (p=1)}	Sibiu (p=1)
{Sibiu (p=3), Timisoara (p=3), Zerind (p=3), Fagaras (p=2), Oradea (p=2), Rimnicu(p=2), Timisoara (p=1), Zerind (p=1)}	Arad (p=2)
{Timisoara (p=3), Zerind (p=3), Fagaras (p=2), Oradea (p=2), Rimnicu(p=2), Timisoara (p=1), Zerind (p=1)}	Sibiu (p=3)
{Zerind (p=3), Fagaras (p=2), Oradea (p=2), Rimnicu(p=2), Timisoara (p=1), Zerind (p=1)}	Timisoara (p=3)
{Fagaras (p=2), Oradea (p=2), Rimnicu(p=2), Timisoara (p=1), Zerind (p=1)}	Zerind (p=3)
{Bucharest (p=3), Sibiu (p=3), Oradea (p=2), Rimnicu(p=2), Timisoara (p=1), Zerind (p=1)}	Fagaras (p=2)
{Sibiu (p=3), Oradea (p=2), Rimnicu(p=2), Timisoara (p=1), Zerind (p=1)}	Bucharest (p=3)

- **Cuestión 2:** Construye el árbol de búsqueda resultante de aplicar el algoritmo **DFS** (en árbol) al problema de búsqueda de una ruta más corta desde Arad a Bucarest con profundidad máxima $m = 3$.

- ▶ **Cuestión 3:** ¿El algoritmo encuentra solución? **Sí**
- ▶ **Cuestión 4:** Si la respuesta es “Sí”:
 - ▷ ¿Cuál ha sido la solución encontrada? **El camino solución encontrado ha sido: Arad, Sibiu, Fagaras, Bucharest**
 - ▷ ¿Cuál es el coste de esta solución? **450**
 - ▷ ¿Se trata de la solución óptima? **No, porque hay otra solución con menor coste de 418: Arad, Sibiu, Rimnicu, Pitesti, Bucharest**
 - ▷ ¿Qué tipo de solución encuentra el algoritmo DFS (en árbol)? **Busca soluciones explorando primero los caminos más profundos (hasta llegar al máximo de profundidad)**
- ▶ **Cuestión 5:** ¿Qué habría ocurrido si no se limita la profundidad máxima? **Que no se hubiese encontrado solución porque se hubiese creado un ciclo entre las ciudades Arad y Sibiu**

Referencias

- [1] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, third edition, 2010.
- [2] Bernhard Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer, 2018.