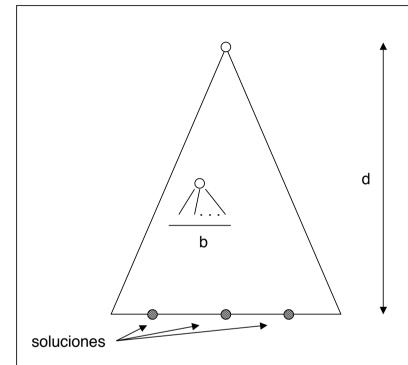
# **Búsqueda Ciega**

- Búsqueda en profundidad
- Búsqueda en anchura
- Profundización iterativa
- Ampliación iterativa

# Búsqueda en Arboles



## Arbol de búsqueda:

- finito: profundidad d
- no hay estados repetidos
- factor de ramificación b uniforme
- soluciones a profundidad d

## Esquema de Búsqueda

#### Esquema:

- L ← lista de nodos iniciales del problema.
   En cualquier momento, L contiene la lista de nodos no visitados.
- Si L vacía, fallo, stop.
  Sino, n ← extrae un nodo de L
- 3. Si *n* es objetivo, éxito, stop. Retornar el camino desde la raíz hasta *n*.
- 4. Sino, eliminar *n* de *L*. <u>Añadir a *L*</u> los <u>sucesores de *n*, etiquetando sus respectivos caminos desde la raíz. Volver al paso 2.</u>

#### Opciones:

extrae un nodo de L

¿al principio? ¿al final?

añadir a *L* 

¿al principio? ¿al final?

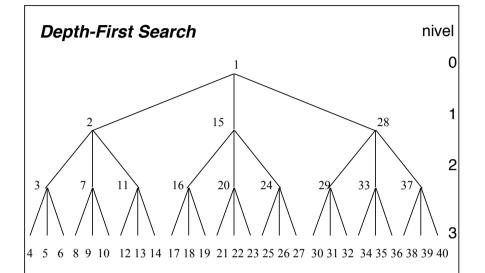
sucesores de n

¿todos? ¿unos pocos?

repetir la búsqueda

otros parámetros

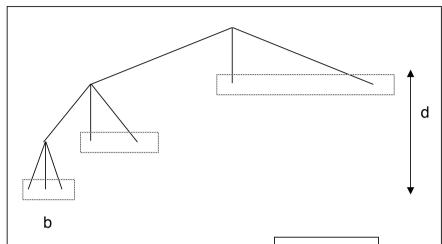
## Búsqueda en Profundidad



#### **Algoritmo DFS:**

- 1. Lista *L* ← nodo raíz
- 2. Si *L* vacía, fallo, stop. Sino, *n* ← extrae-primero(*L*).
- 3. Si *n* es objetivo, éxito, stop. Sino, generar los sucesores de *n*.
- 4. Añadir <u>al principio</u> de *L* todos los sucesores de *n*, etiquetando cada sucesor con su camino desde la raíz. Ir al paso 2.

## Búsqueda en Profundidad



Espacio:

$$d(b-1)+1$$

lineal en d

#### Tiempo: (nodos visitados)

objetivo: hoja más a la izquierda

$$d + 1$$

objetivo: hoja más a la derecha

$$1 + b + b^2 + \dots + b^d = \frac{b^{d+1} - 1}{b - 1}$$

media:

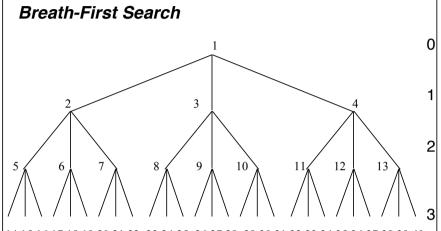
$$\frac{b^{d+1} + bd + b - d - 2}{2(b-1)}$$

 $= \frac{b^{d+1}}{2(b-1)}$  (d grande)

exponencial en d



## Búsqueda en Anchura

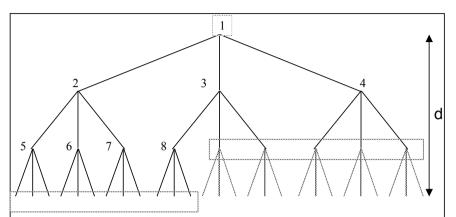


14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40

### **Algoritmo BFS:**

- 1. Lista *L* ← nodo raíz
- 2. Si *L* vacía, fallo, stop. Sino, *n* ← extrae-primero(*L*).
- 3. Si *n* es objetivo, éxito, stop. Sino, generar los sucesores de *n*.
- 4. Añadir <u>al final</u> de *L* todos los sucesores de *n*, etiquetando cada sucesor con su camino desde la raíz. Ir al paso 2.

# Búsqueda en Anchura



**Espacio:** 

bd

exponencial en d

#### Tiempo: (nodos visitados)

nodos internos

$$1 + b + b^2 + \dots + b^{d-1} = \frac{b^d - 1}{b - 1}$$

media nodos hoja

$$\frac{1+b^d}{2}$$



media nodos visitados

$$\frac{b^{d+1} + b^d + b - 3}{2(b-1)} \quad \approx \quad \frac{b^{d+1} + b^d}{2(b-1)} \tag{d grande}$$

*exponencial* en d

## **Profundidad vs Anchura**

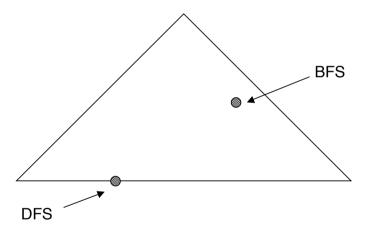
Espacio: lineal vs exponencial

Tiempo: ambos exponenciales

DFS mejor que BFS por un factor

$$1 + \frac{1}{b}$$

#### Características:



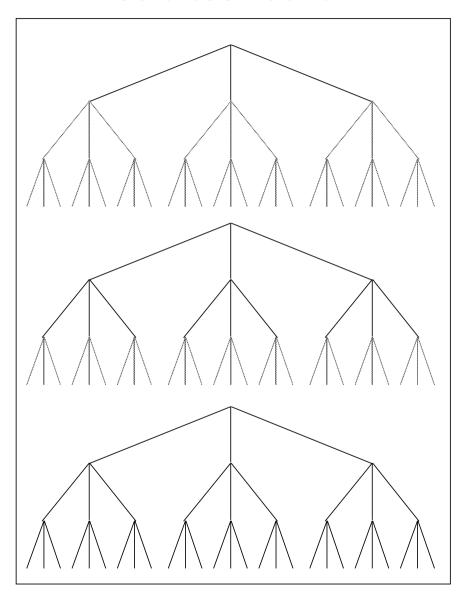
Solución más cercana a la raíz:

BFS: Sí

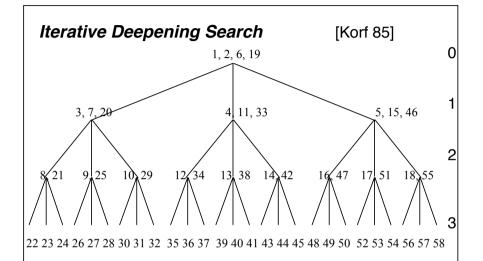
DFS: No

DFS y ramas infinitas?

## Profundización Iterativa



## Profundización Iterativa



### Algoritmo:

- 1. Cota profundidad  $c \leftarrow 1$ .
- 2. Lista  $L \leftarrow \text{nodo raíz}$ .
- 3. Si L vacia,  $c \leftarrow c + 1$ , ir al paso 2. Sino,  $n \leftarrow extrae-primero(L)$ .
- 4. Si *n* es objetivo, éxito, stop.
- 5. Si *profundidad*(*n*) < c, añadir <u>al principio</u> de *L* todos los sucesores de *n*, etiquetando cada sucesor con su camino desde la raíz. En cualquier caso, ir al paso 3.

## Profundización Iterativa

#### Espacio: igual que DFS

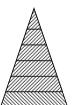
$$d(b-1)+1$$

*lineal* en d

#### Tiempo: (nodos visitados)

iteraciones anteriores: a profundidad j

$$\sum_{i=0}^{d-1} \frac{b^{i+1}-1}{b-1} = \frac{b^{d+1}-b^d-b+d}{(b-1)^2}$$



última iteración: a profundidad d

$$\frac{b^{d+1} + bd + b - d - 2}{2(b - 1)}$$

sumando

$$\frac{b^{d+2} + b^{d+1} + b^2d + b^2 - 4bd - 5b + 3d + 2}{2(b-1)^2}$$

$$\approx \frac{b^{d+2} + b^{d+1}}{2(b-1)^2}$$
 (d grande)

exponencial en d

#### DFS / BFS vs IDA

DFS

**BFS** 

**IDA** 

Espacio:

lineal

exp

lineal

Tiempo:

exp

exp

exp

$$\frac{BFS}{DFS} = \frac{b+3}{b}$$

$$\frac{\text{IDA}}{\text{BFS}} = \frac{\text{b}}{\text{b-1}}$$

$$\frac{\text{IDA}}{\text{DFS}} = \frac{b+1}{b-1}$$

#### Características:

Solución más cercana a la raíz

BFS: Sí

DFS: No

IDA: Sí

Problema con ramas infinitas

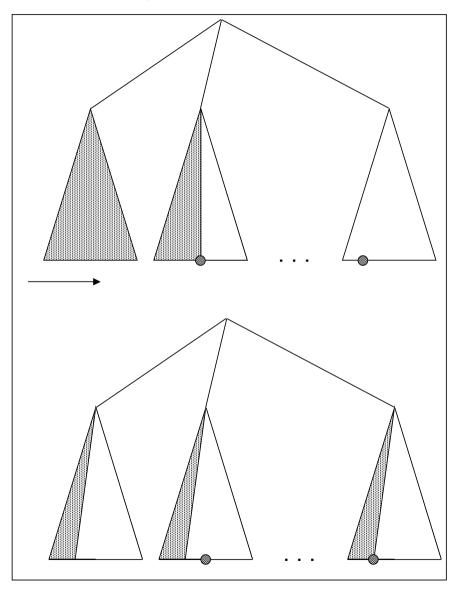
BFS: No

DFS: Sí

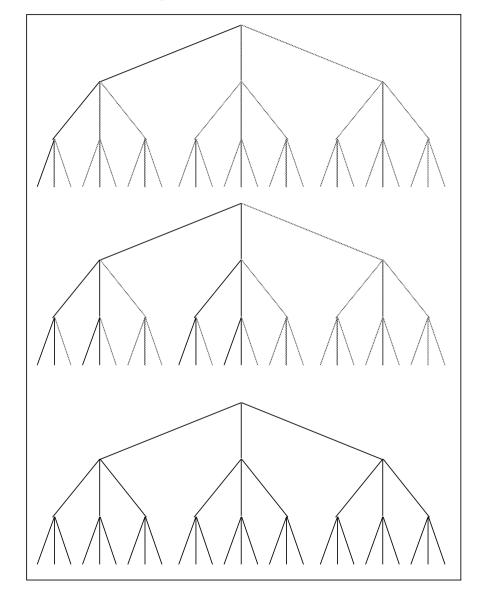
IDA: No

Solución a profundidad fija: DFS

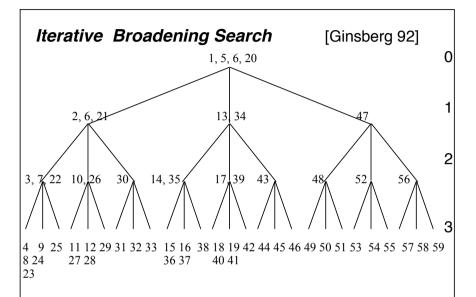
# Ampliación Iterativa



# Ampliación Iterativa



## **Ampliación Iterativa**



#### **Algoritmo:**

- 1. Cota anchura  $c \leftarrow 1$ .
- 2. Lista  $L \leftarrow \text{nodo raíz}$ .
- 3. Si L vacía,  $c \leftarrow c + 1$ , ir al paso 2. Sino,  $n \leftarrow extrae-primero(L)$ .
- 4. Si n es objetivo, éxito, stop.
- 5. Añadir <u>al principio</u> de *L* los primeros *c* sucesores de *n*, etiquetando cada sucesor con su camino desde la raíz. Ir al paso 3.

## **Ampliación Iterativa**

Espacio: igual que DFS

d (b - 1) + 1 *lineal* en d

Tiempo: (nodos visitados) ≈

$$d + \quad 2^d \quad + \ 3^d + \ldots + \ b^d$$

$$\frac{b^{d+1}}{d}$$
 (b grande)