Inteligencia Computacional

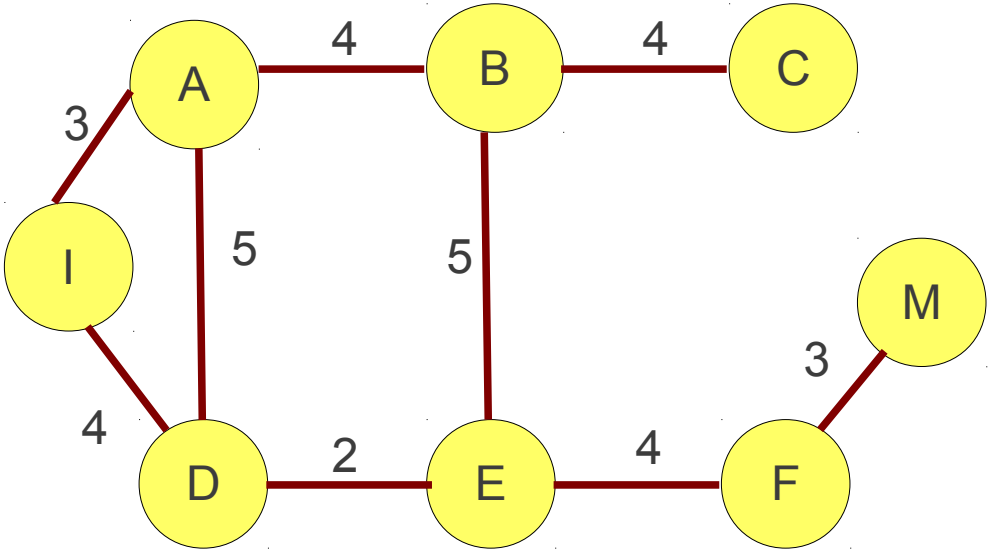
Búsqueda y optimización ciega

http://blancavg.com/tc3023/

Blanca A. Vargas Govea * blanca.vg@gmail.com * Agosto 14, 2012

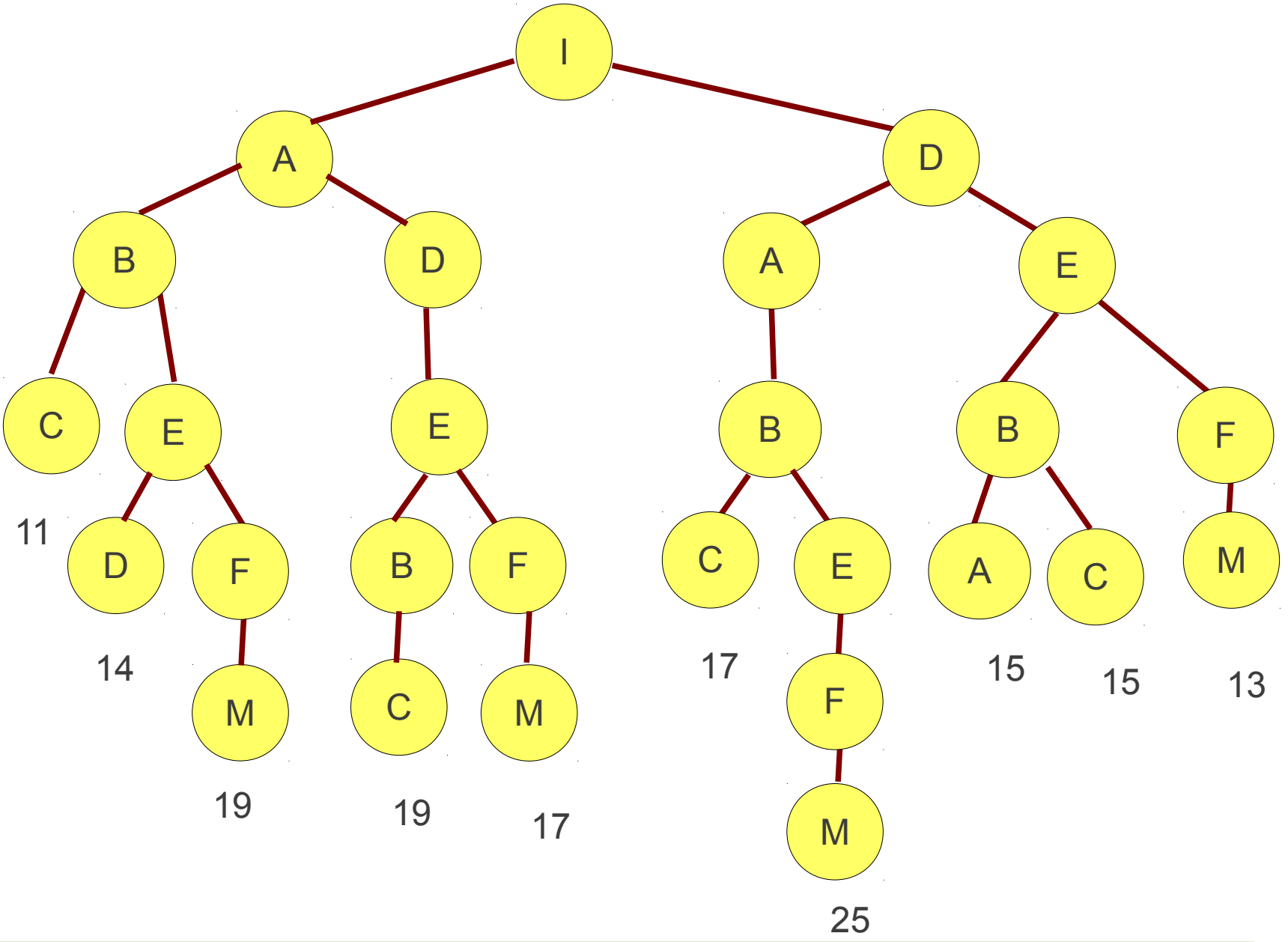
Búsqueda de soluciones

Para resolver los problemas se requiere de una búsqueda sobre el espacio de estados.



Grafo

Representación



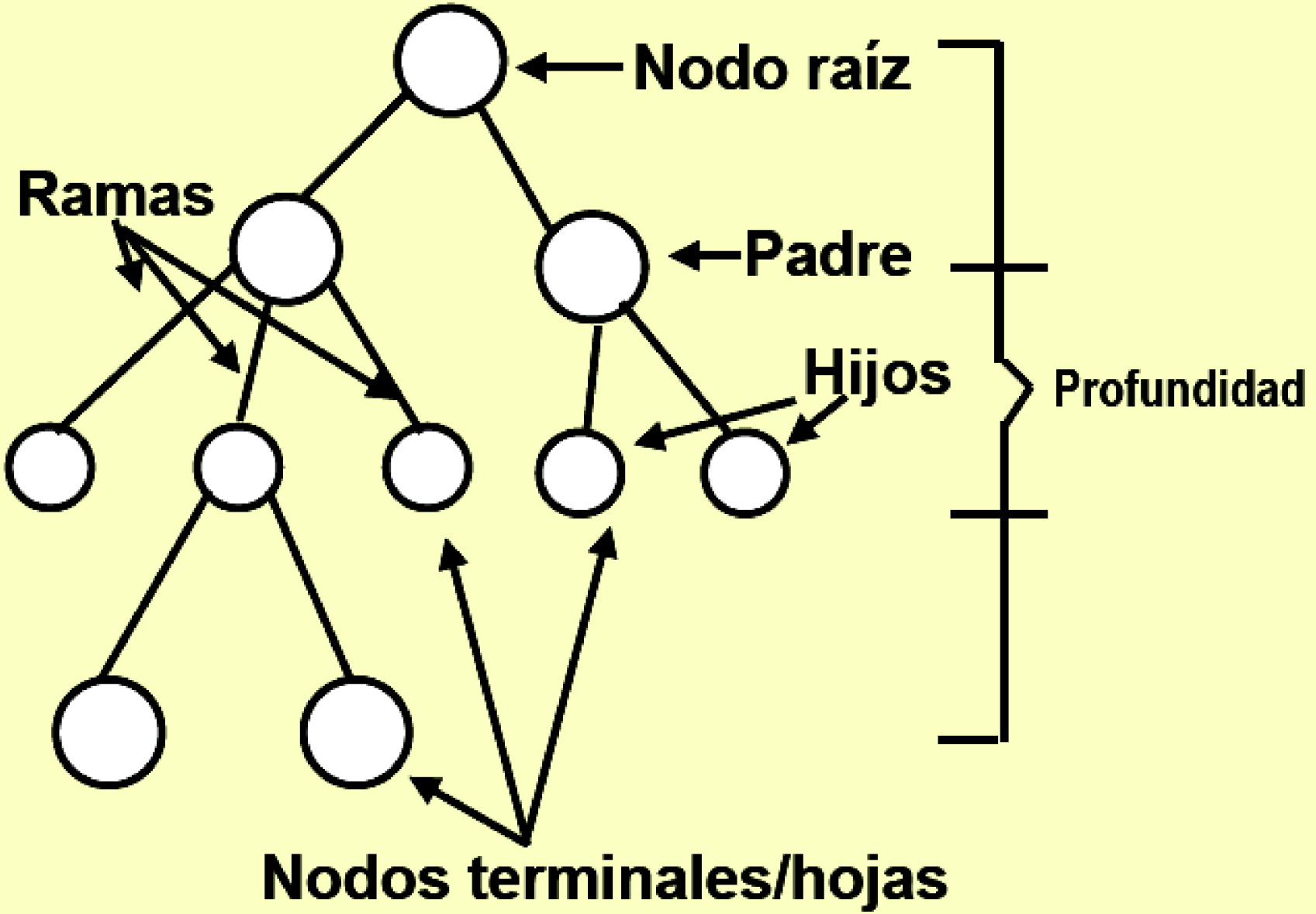
Representación - árbol

1. Estructura simbólica que represente conjuntos de soluciones potenciales (agenda).
2. Operaciones ó reglas que modifiquen símbolos de la agenda y produzcan conjuntos de soluciones potenciales más refinadas.
3. Estrategia de búsqueda para decidir qué operación aplicar a la agenda.

Representación

1. Estructura simbólica que represente conjuntos de soluciones potenciales (agenda).
2. Operaciones ó reglas que modifiquen símbolos de la agenda y produzcan conjuntos de soluciones potenciales más refinadas.
3. Estrategia de búsqueda para decidir qué operación aplicar a la agenda.

Representación


Nodo raíz
Ramas
Padre
Hijos
Profundidad
Nodos terminales/hojas

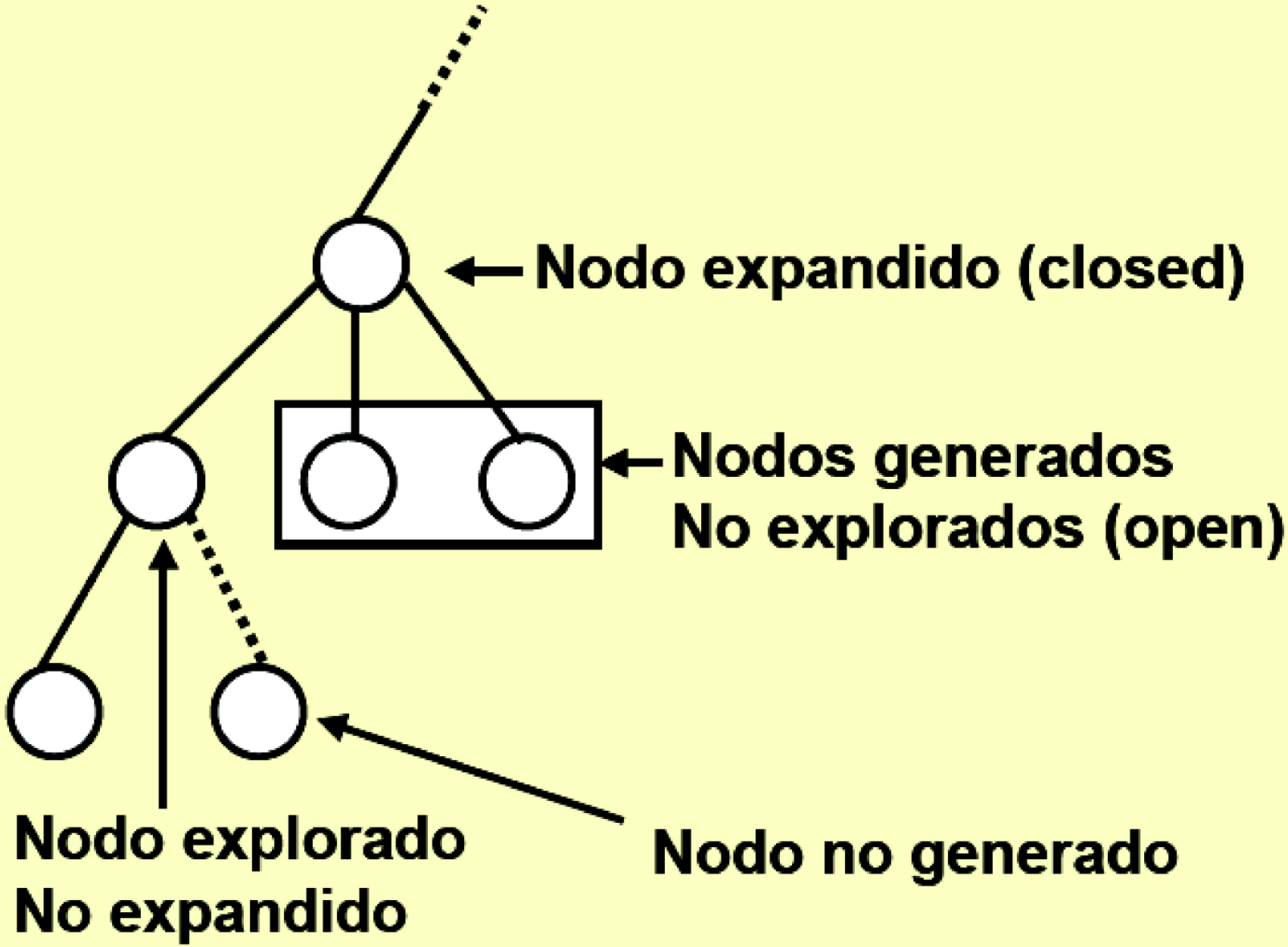
Expansión

Aplicar la función-sucesores al estado actual generando un nuevo conjunto de estados

Estrategia de búsqueda

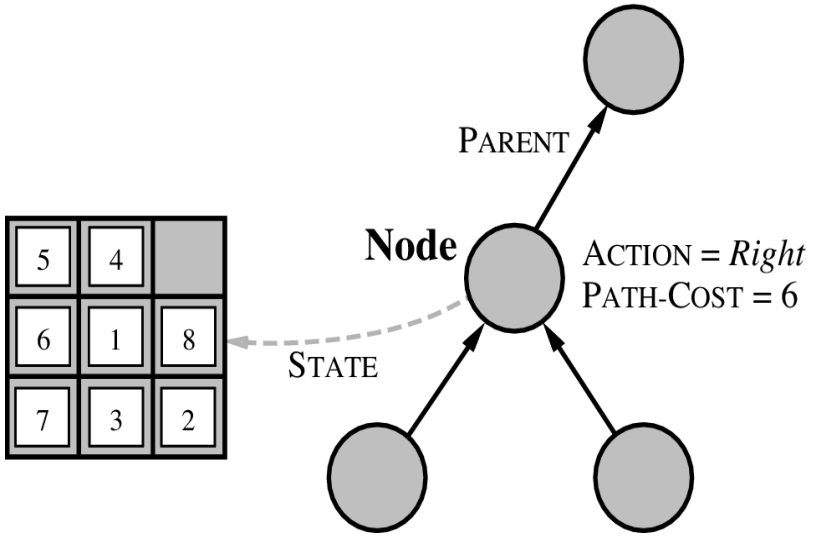
Decisión sobre qué nodo expandir

Terminología


Nodo expandido (closed)
Nodos generados
No explorados (open)
Nodo explorado
No expandido
Nodo no generado

Un estado es una representación de una configuración física

Un nodo es una estructura de datos, parte de una búsqueda incluye padre, hijos, profundidad, costo



Nodos vs Estados

```
function TREE-SEARCH( problem, strategy) returns a solution, or failure
    initialize the search tree using the initial state of problem
    loop do
        if there are no candidates for expansion then return failure
        choose a leaf node for expansion according to strategy
        if the node contains a goal state then return the corresponding solution
        else expand the node and add the resulting nodes to the search tree
    end
```

Algoritmo informal

Completez

¿El algoritmo garantiza encontrar una solución?

¿Es óptimo?

¿La estrategia encuentra la solución óptima?

Tiempo

¿Cuánto tarda en encontrar una solución?

Espacio

¿Cuánta memoria se necesita para efectuar la búsqueda?

Evaluación de estrategias

La estrategia es sistemática si:

1. No deja un solo camino sin explorar (completo)

2. No explora un mismo camino más de una vez (eficiencia)

Propiedades

Sin información (ciega)

Solamente puede distinguir entre un estado meta y uno no-meta.
Algún camino: breadth-first, depth-first

Con información (heurística)

Pueden saber si un estado no-meta es más prometedor que otro.
Algún camino: hill climbing, beam search, best first

Óptimo

British museum, branch and bound, A*

Tipos de estrategia

No existe información adicional más allá de la definición del problema.

Lo único que pueden hacer es generar sucesores y distinguir entre un estado meta y no-meta.

Estrategias sin información (búsqueda ciega)

```
Crea una agenda de un elemento (el nodo raíz)
hasta que la agenda esté vacía o se alcance la meta
    si el primer elemento es la meta
    entonces acaba
    si no elimina el primer elemento y
        añade sus sucesores al final de la agenda
```

Todos los nodos de un nivel son expandidos antes de pasar a otro nivel.

Es exhaustiva (expande todos los nodos).

Algoritmo: búsqueda a lo ancho

Lo bueno

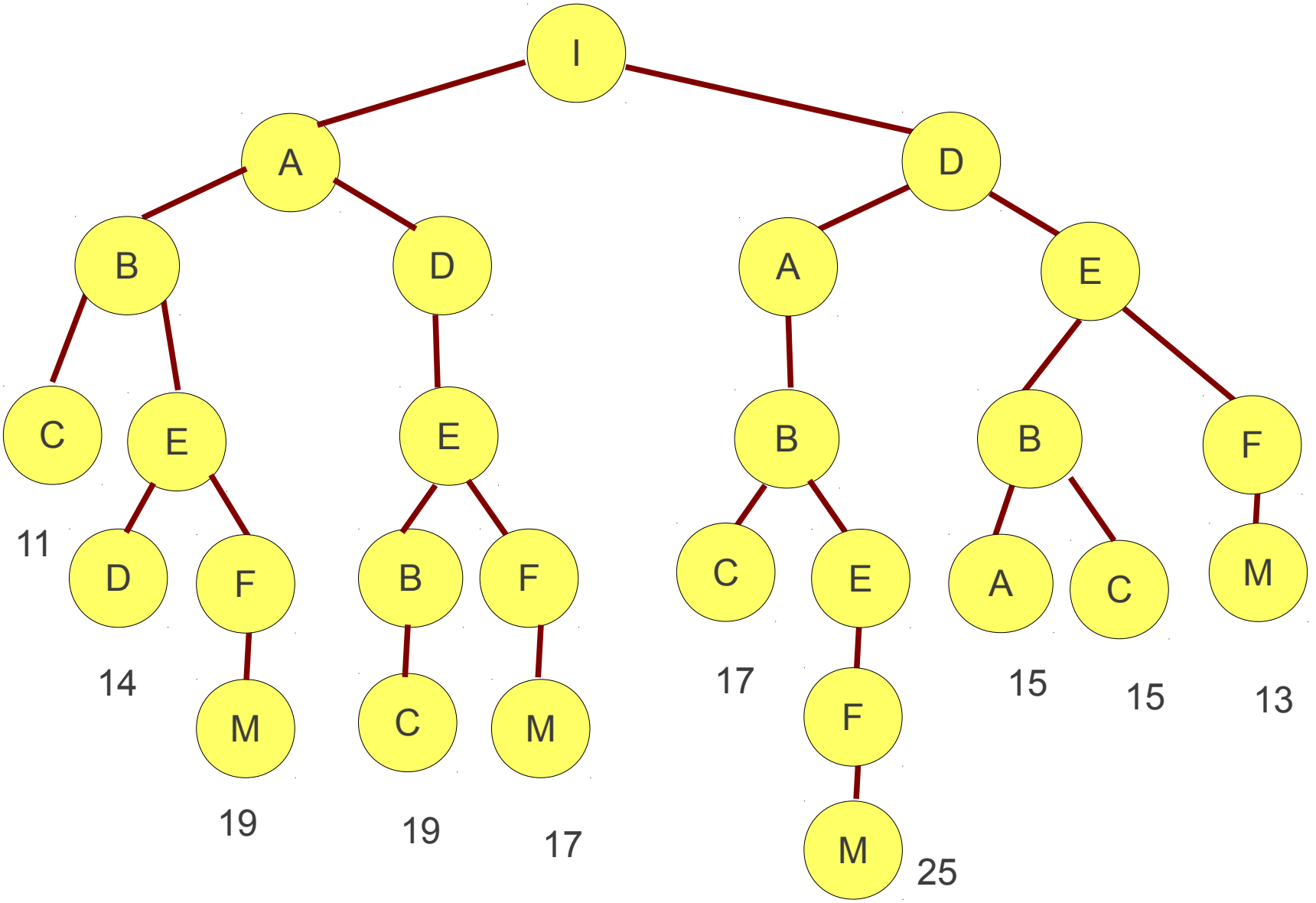
Es completo: encuentra una solución si existe

Es óptimo: si el costo del camino no decrece con la profundidad del nodo (e.g., costos iguales).

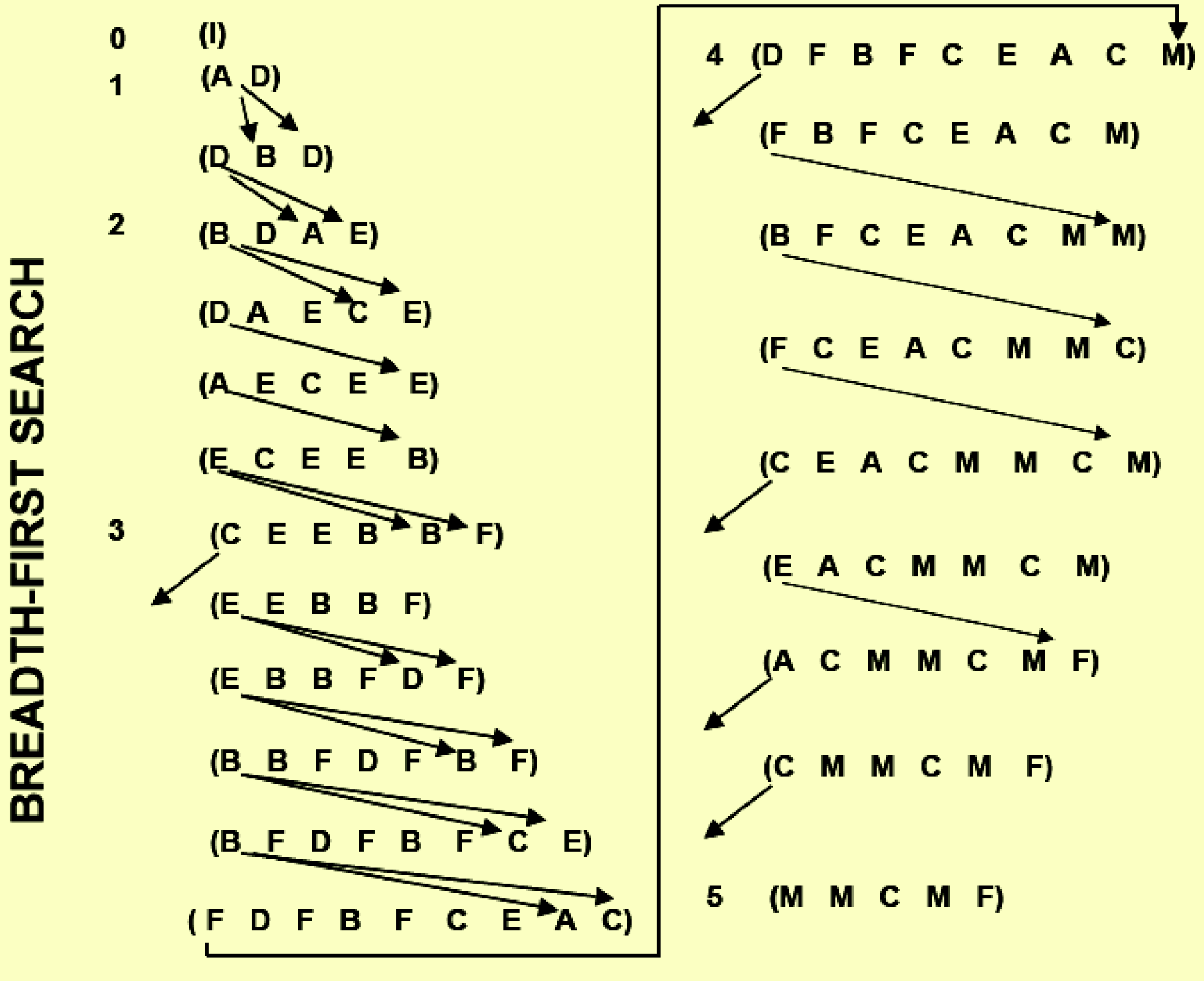
Lo malo

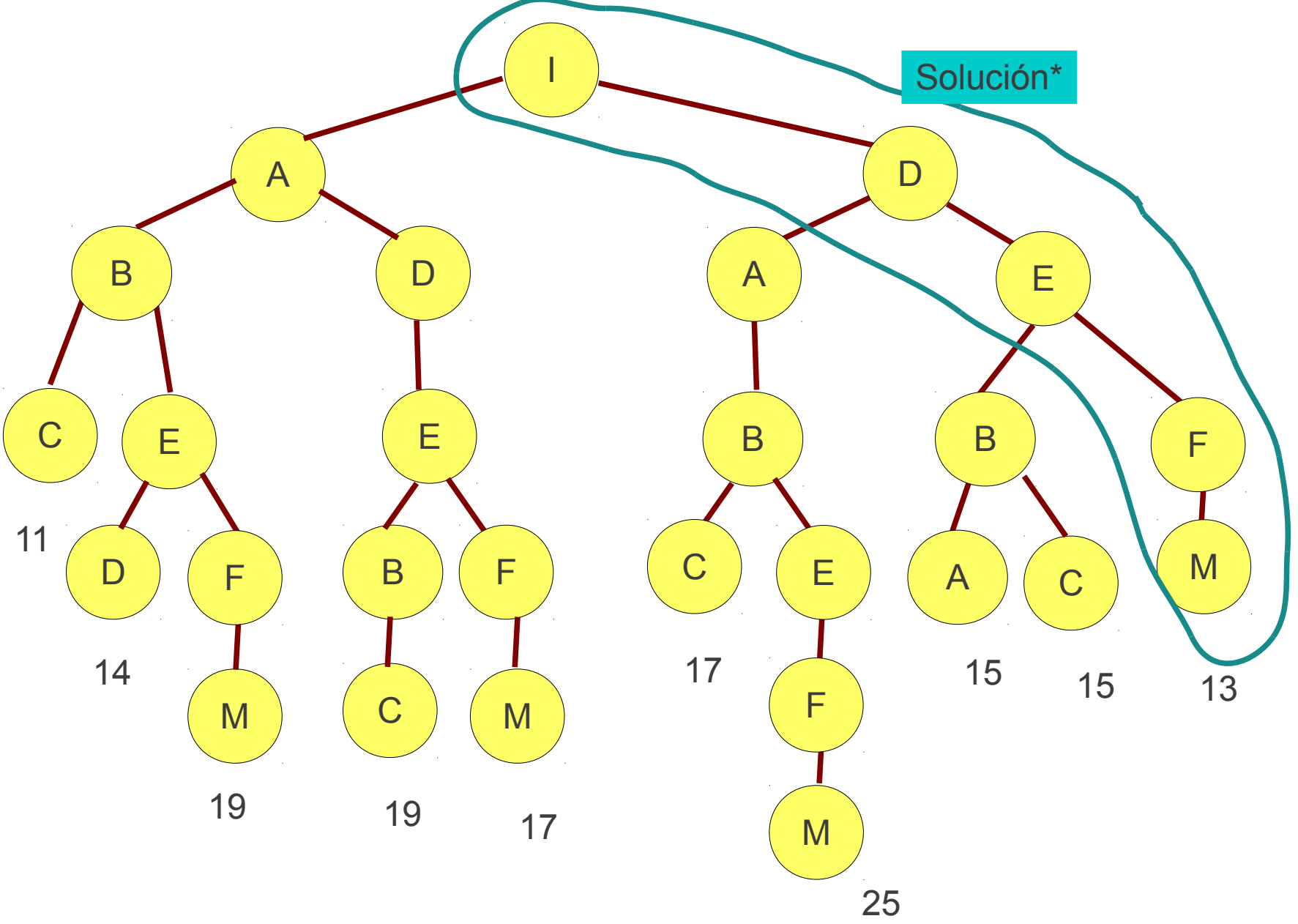
Alto costo computacional.

Algoritmo: búsqueda a lo ancho



Algoritmo: búsqueda a lo ancho


BREADTH-FIRST SEARCH
0 (I)
1 (A D)
(D B D)
2 (B D A E)
(D A E C E)
(A E C E E)
(E C E E B)
3 (C E E B B F)
(E E B B F)
(E B B F D F)
(B B F D F B F)
(B F D F B F C E)
(F D F B F C E A C)
4 (D F B F C E A C M)
(F B F C E A C M)
(B F C E A C M M)
(F C E A C M M C)
(C E A C M M C M)
(E A C M M C M)
(A C M M C M F)
(C M M C M F)
5 (M M C M F)



Algoritmo: búsqueda a lo ancho

```
Crea una agenda de un elemento (el nodo raíz)
hasta que la agenda esté vacía o se alcance la meta
    si el primer elemento es la meta
    entonces acaba
    si no elimina el primer elemento y
        añade sus sucesores al frente de la agenda
```

Cada nodo que es explorado genera todos sus sucesores antes de que otro nodo sea explorado.

Backtracking - regresa al nodo más reciente que no ha sido explorado.

Algoritmo: búsqueda en profundidad

Lo bueno

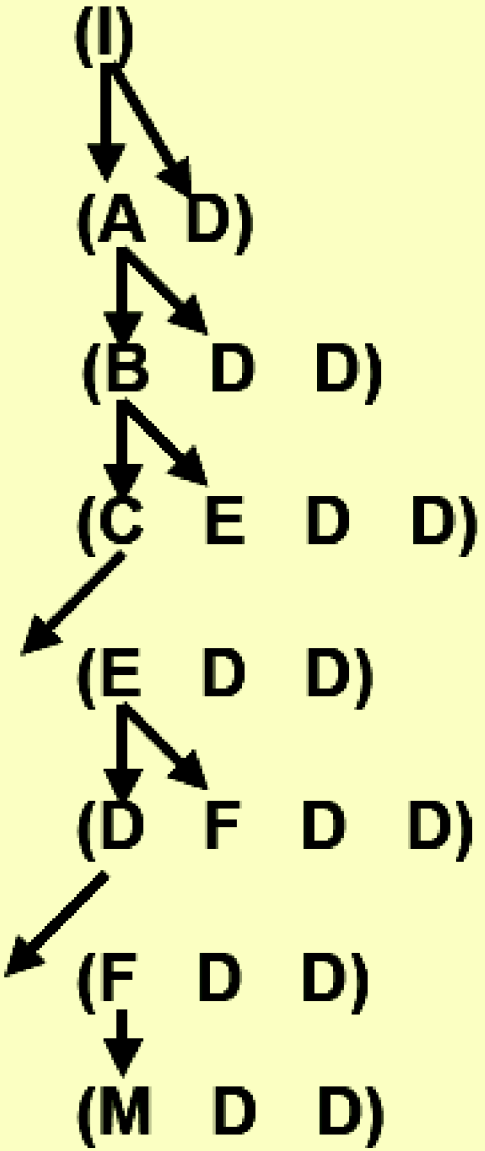
Tiene pocos requerimientos de memoria. Solamente guarda un camino desde la raíz al nodo hoja y se elimina cuando fue explorado.

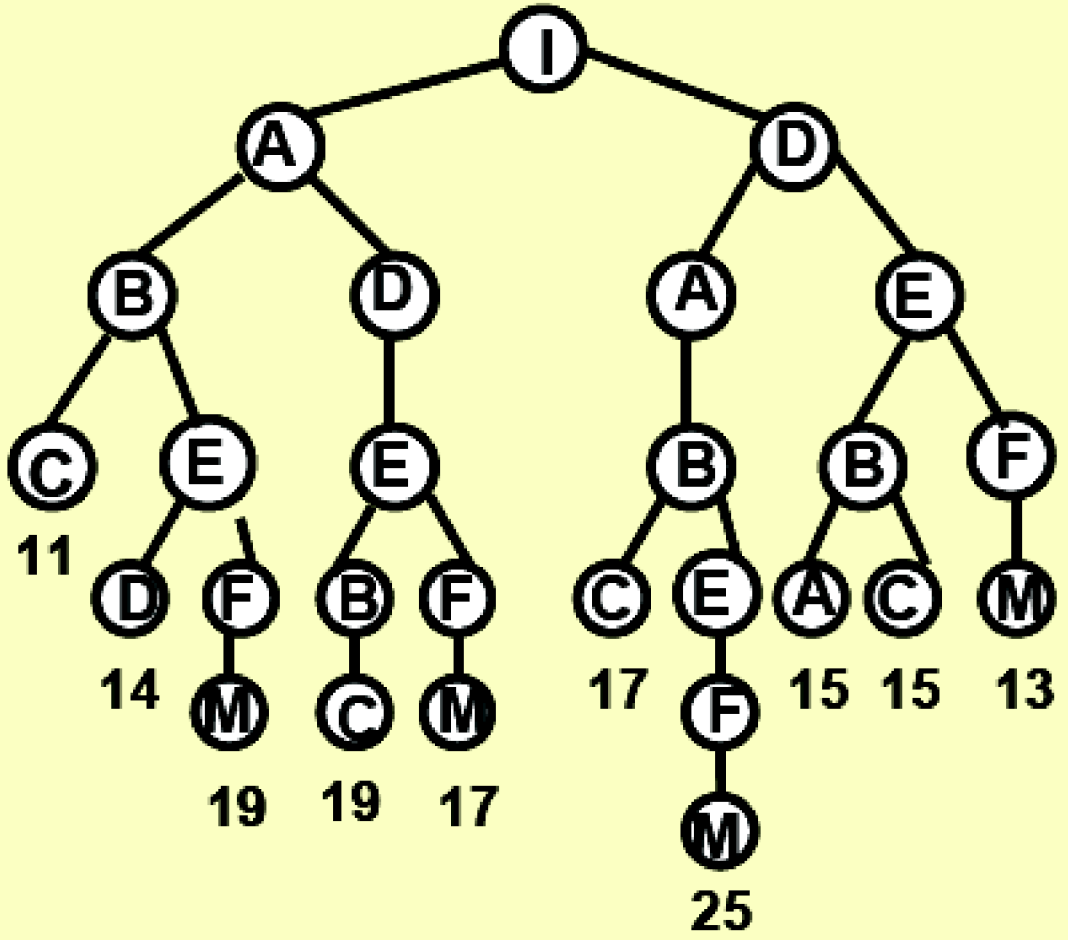
Lo malo

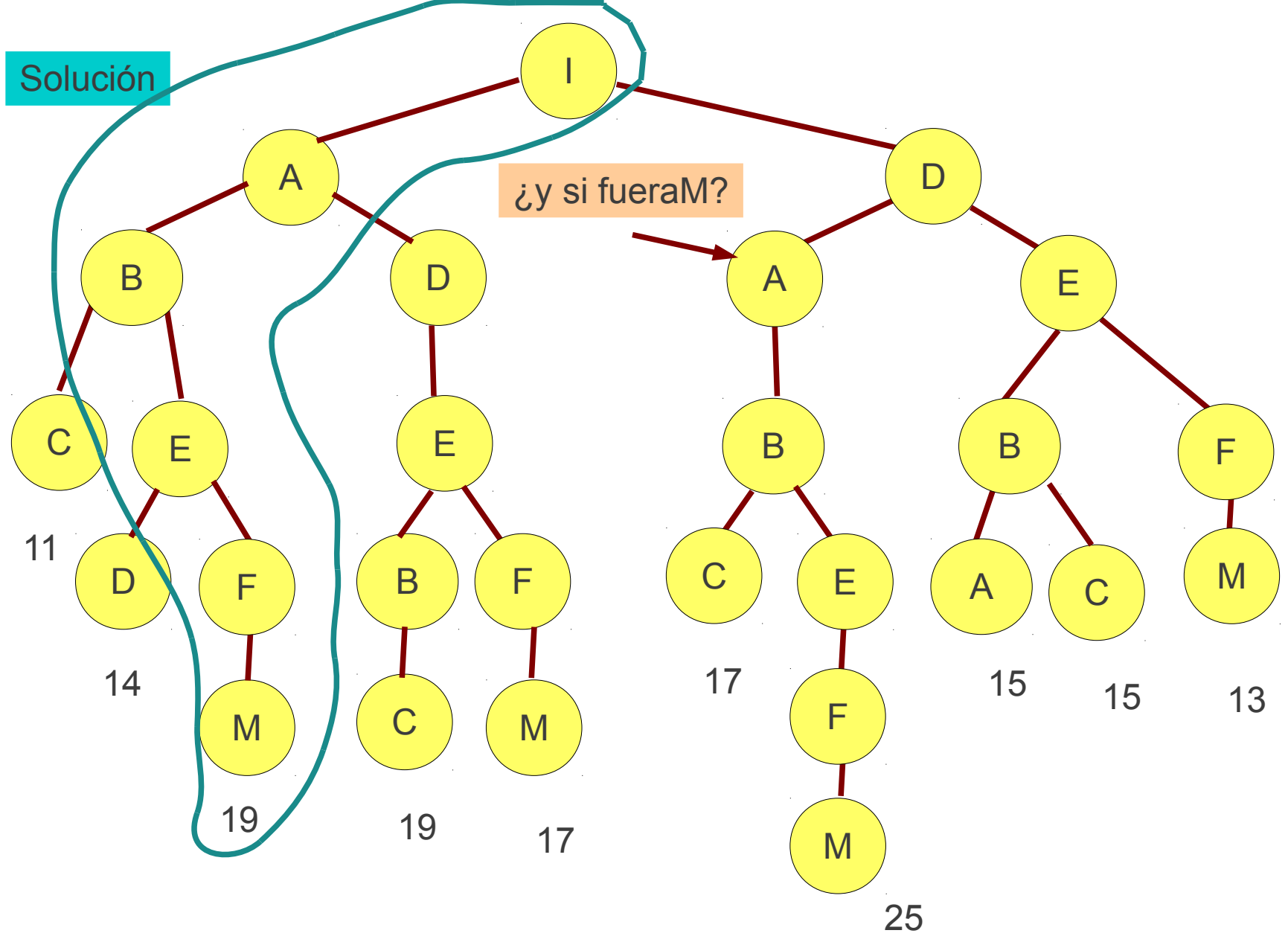
No es óptimo.

Algoritmo: búsqueda en profundidad

DEPTH-FIRST SEARCH







Algoritmo: búsqueda en profundidad

Dada una función objetivo, la optimización ciega es la búsqueda de parámetros para minimizar o maximizar la función sin tener acceso a la función. Lo único que se permite saber es el resultado de la evaluación.

Búsqueda aleatoria: el nodo a expander se selecciona aleatoriamente. Se itera hasta cumplir con un criterio de terminación (no. de intentos, tiempo de ejecución, umbral de aceptación).

Optimización ciega

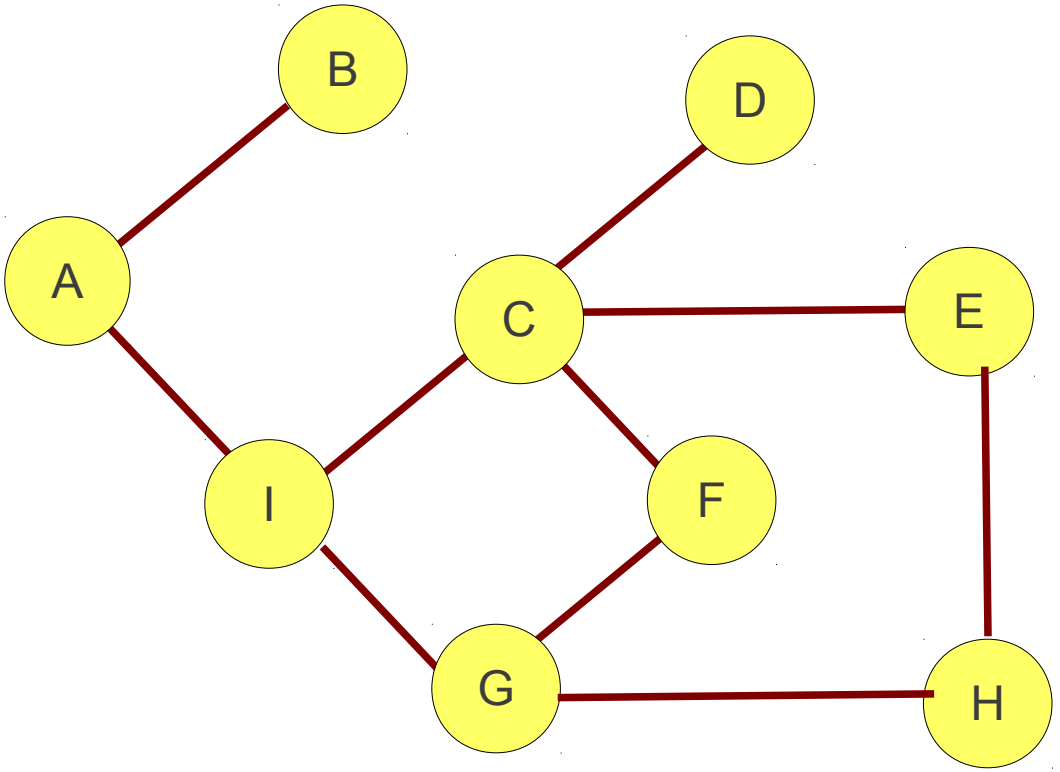
Los problemas de optimización combinatoria son aquellos en los que las soluciones son codificadas con variables discretas. Consiste en encontrar el orden de un conjunto de variables que ayuden a maximizar o minimizar el valor de la función objetivo.

Optimización combinatoria

Opciones:

- No regresar al estado anterior.
- No crear caminos con ciclos.
- No generar algún estado generado antes.

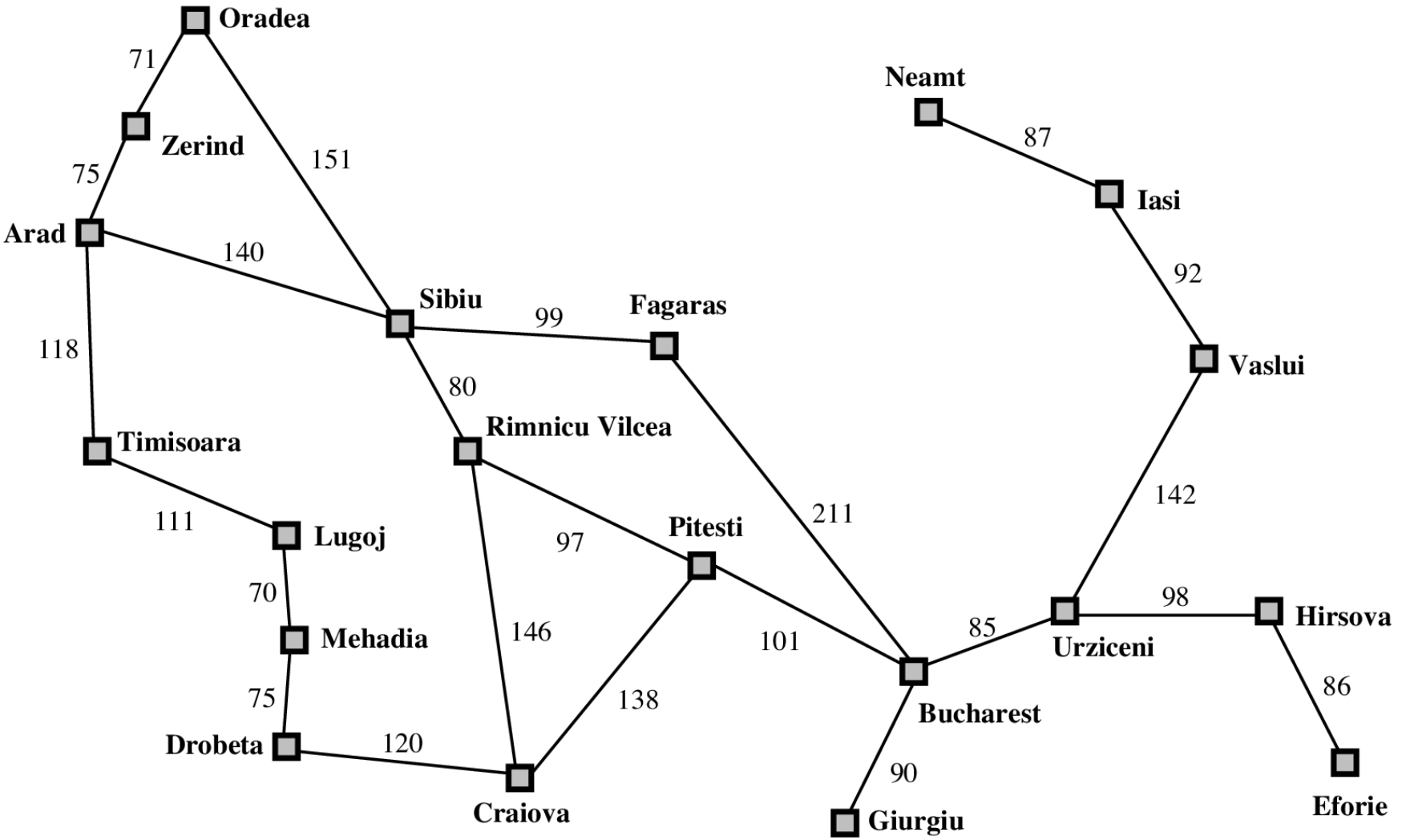
Estados repetidos



1. Hacer el recorrido en anchura.
2. Hacer el recorrido en profundidad.

Para ambos, escribe los nodos explorados y la agenda (ver ejemplos). Entregar.

Tarea



Implementar la búsqueda a lo ancho y en profundidad y encontrar las rutas óptimas:
1) Arad – Bucharest
2) Dobreta – Fagaras
Entregar el código fuente y ejecutable y un breve reporte en el que analices las diferencias entre los resultados dados por los algoritmos con base en: ruta óptima, completez y tiempo.
Entrega: Martes 21 de Agosto

Referencias

Russell, S., y Norvig, P. (2003). Artificial intelligence: A modern approach (2nd edition ed.). Prentice-Hall, Englewood Cliffs, NJ.

Blind http://www.flickr.com/photos/24368739@N00/2098529830/
Notas de cursos: Eduardo Morales, L. Enrique Sucar