

LABORATORIO 03

A* SEARCH

Docentes: Edward Hinojosa Cárdenas

12 de Mayo del 2023

1 CONCEPTOS BÁSICOS

- Búsqueda Informada
- A* Search

2 EQUIPOS Y MATERIALES

- Un computador.
- Material del curso.

3 EJERCICIOS

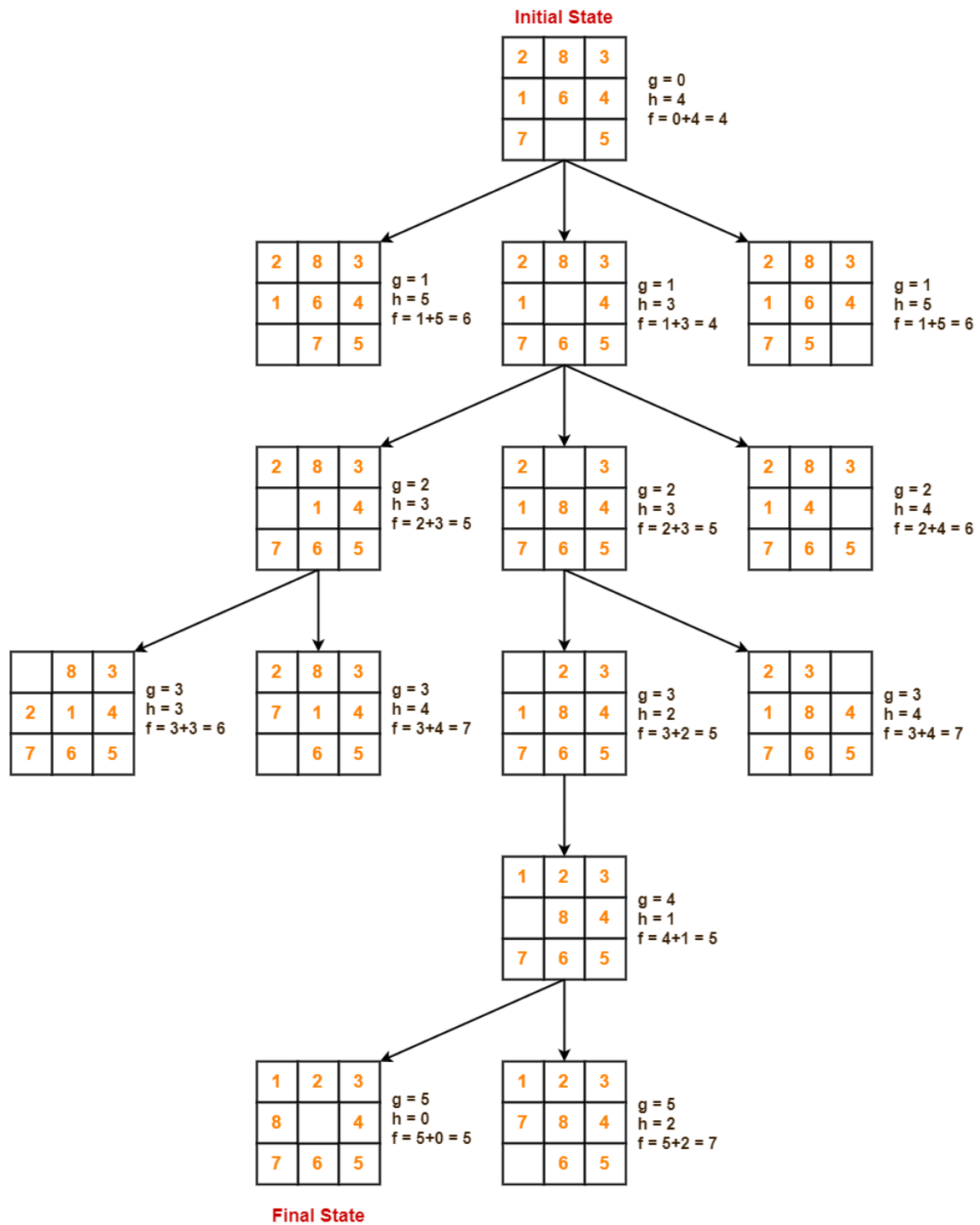
Considere el siguiente grafo:

1. Implemente (en cualquier lenguaje de programación) la A* search desde un estado inicial a un estado final en el juego del Rompecabezas de 8 piezas (el que se mostró en clases). Muestre la búsqueda paso a paso (o iteración por iteración) (mostrando la lista *Open* y *Closed* correspondiente). El archivo .txt debe comenzar con el nombre del alumno, nombre de la búsqueda, estado inicial y estado final. Debe mostrar una ejecución con más de 6 pasos. (16 puntos)
2. Muestre la solución anterior de forma gráfica. (4 puntos)

4 ALGORITMO A* SEARCH

```
// A*
1: initialize the open list
2: initialize the closed list
3: put the starting node on the open list (you can leave its f at zero)
-
4: while the open list is not empty
5:   find the node with the least f on the open list, call it "q"
6:   pop q off the open list
7:   generate q successors and set their parents to q
8:   for each successor
9:     if successor is the goal, stop the search
10:    successor.g = q.g + distance between successor and q
11:    successor.h = distance from goal to successor
12:    successor.f = successor.g + successor.h
-
13:    if a node with the same position as successor is in the OPEN list \
-      which has a lower f than successor, skip this successor
14:    if a node with the same position as successor is in the CLOSED list \
-      which has a lower f than successor, skip this successor
15:    otherwise, add the node to the open list
16:   end
17:   push q on the closed list
18: end
```

5 JUEGO DEL ROMPECABEZA DE 8 PIEZAS



6 EJEMPLO

Nombre del alumno Search A*

Estado Inicial [2, 8, 3, 1, 6, 4, 7, , 5]

Estado Final [1, 2, 3, 8, , 4, 7, 6, 5]

Open: [2, 8, 3, 1, 6, 4, 7, , 5]; g=0; h=4; f=4 Closed:

Iteración 1 Open: [2, 8, 3, 1, , 4, 7, 6, 5]; g=1; h=3; f=4 [2, 8, 3, 1, 6, 4, , 7, 5]; g=1; h=5; f=6 [2, 8, 3, 1, 6, 4, 7, 5,]; g=1; h=5; f=6 Closed: [2, 8, 3, 1, 6, 4, 7, , 5]; g=0; h=4; f=4

Iteración 2 Open: [2, 8, 3, , 1, 4, 7, 6, 5]; g=2; h=3; f=5 [2, , 3, 1, 8, 4, 7, 6, 5]; g=2; h=3; f=5 [2, 8, 3, 1, 6, 4, , 7, 5]; g=1; h=5; f=6 [2, 8, 3, 1, 6, 4, 7, 5,]; g=1; h=5; f=6 [2, 8, 3, 1, 4, , 7, 6, 5]; g=2; h=4; f=6 Closed: [2, 8, 3, 1, 6, 4, 7, , 5]; g=0; h=4; f=4 [2, 8, 3, 1, , 4, 7, 6, 5]; g=1; h=3; f=4

Iteración 3 Open: [2, , 3, 1, 8, 4, 7, 6, 5]; g=2; h=3; f=5 [2, 8, 3, 1, 6, 4, , 7, 5]; g=1; h=5; f=6 [2, 8, 3, 1, 6, 4, 7, 5,]; g=1; h=5; f=6 [2, 8, 3, 1, 4, , 7, 6, 5]; g=2; h=4; f=6 [, 8, 3, 2, 1, 4, 7, 6, 5]; g=3; h=3; f=6 [2, 8, 3, 7, 1, 4, , 6, 5]; g=3; h=4; f=7 Closed: [2, 8, 3, 1, 6, 4, 7, , 5]; g=0; h=4; f=4 [2, 8, 3, 1, , 4, 7, 6, 5]; g=1; h=3; f=4 [2, 8, 3, , 1, 4, 7, 6, 5]; g=2; h=3; f=5

Iteración 4 Open: [, 2, 3, 1, 8, 4, 7, 6, 5]; g=3; h=2; f=5 [2, 8, 3, 1, 6, 4, , 7, 5]; g=1; h=5; f=6 [2, 8, 3, 1, 6, 4, 7, 5,]; g=1; h=5; f=6 [2, 8, 3, 1, 4, , 7, 6, 5]; g=2; h=4; f=6 [, 8, 3, 2, 1, 4, 7, 6, 5]; g=3; h=3; f=6 [2, 8, 3, 7, 1, 4, , 6, 5]; g=3; h=4; f=7 [2, 3, , 1, 8, 4, 7, 6, 5]; g=3; h=4; f=7 Closed: [2, 8, 3, 1, 6, 4, 7, , 5]; g=0; h=4; f=4 [2, 8, 3, 1, , 4, 7, 6, 5]; g=1; h=3; f=4 [2, 8, 3, , 1, 4, 7, 6, 5]; g=2; h=3; f=5 [2, , 3, 1, 8, 4, 7, 6, 5]; g=2; h=3; f=5

7 ENTREGABLES

Al finalizar el estudiante deberá:

1. Generar los archivos .txt solicitados.
2. Comprimir en un archivo .zip todos los archivos anteriores (además de todo código fuente, sin el cual no se revisará el laboratorio) y subirlo el archivo al aula virtual cuando indique el profesor:

Laboratorio_XX_ApellidoPaterno_ApellidoMaterno_PrimerNombre_IA_2023A_EPCC_UNSA.zip