



ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA EN SISTEMAS  
INGENIERÍA EN COMPUTACIÓN



PERÍODO ACADÉMICO: 2020 A  
ASIGNATURA: Inteligencia Artificial  
PROFESOR: Dra. Myriam Hernández  
FECHA DE ENTREGA: 26/06/2020  
NOMBRE ESTUDIANTE: Edison Daniel Cabrera Pabón

### Laboratorio 3 Implementación Heurística en 8 Puzzle

Programa Implementado en Python:

```
13 # se estableció una clase NODO
14 class Node(object):
15     def __init__(self, data, prev, next):
16         self.data = data
17         self.prev = prev
18         self.next = next
19
20 # se estableció una LISTA DOBLEMENTE ENLAZADA
21 # con 3 métodos para la correcta asignación de las posiciones de las reinas
22 class DoubleList(object):
23     head = None
24     tail = None
25     best = None
26     explorado = 1
27
28     def append(self, data):
29         new_node = Node(data, None, None)
30         if self.head is None:
31             self.head = self.tail = self.best = new_node
32         else:
33             new_node.prev = self.best
34             new_node.next = None
35             self.tail.next = new_node
36             self.tail = new_node
37
38     def recover(self):
39         current_node = self.best
40         while current_node is not None:
41             yield current_node.data
42             current_node = current_node.prev
43
44     def bestNode(self):
45         f = 0
46         m = 2000
47         h = 0
48         nm = m
49
50         i = self.head
51         while i is not None:
52             f = i.data.h + i.data.g
53             if(i.data.explorado==0 and f < m):
54                 m = f
55             i = i.next
56
57         i = self.head
58         while i is not None:
59             f = i.data.h + i.data.g
60             h = i.data.h
61             if(i.data.explorado==0 and f==m and h < nm):
62                 self.best = i
63                 nm = h
64             i = i.next
65
66         self.explorado = self.explorado + 1
67         self.best.data.explorado = self.explorado
68
```



ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA EN SISTEMAS  
INGENIERÍA EN COMPUTACIÓN



```
69
70 # se estableció el ESCENARIO DE JUEGO
71 class Puzzle(object):
72     coordenadas_x = [1,1,1,2,2,2,3,3,3]
73     coordenadas_y = [1,2,3,1,2,3,1,2,3]
74     elementos      = ["1","2","3","4","5","6","7","8"]
75     intercambios    = [[1,3,9,9],
76                        [0,2,4,9],
77                        [1,5,9,9],
78                        [0,4,6,9],
79                        [1,3,5,7,9],
80                        [2,4,8,9],
81                        [3,7,9,9],
82                        [4,6,8,9],
83                        [5,7,9,9]]
84
85     def __init__(self):
86         self.estado = None
87         self.h = 0
88         self.g = 0
89         self.explorado = 0
90         self.ascendiente = None
91
92     def posx(self,c):
93         return self.coordenadas_x[self.estado.index(c)]
94
95     def posy(self,c):
96         return self.coordenadas_y[self.estado.index(c)]
```

```
98     def distanciaManhatan(self,meta):
99         suma = 0
100         for j in self.elementos:
101             dmx = abs(self.posx(j) - meta.posx(j))
102             dmy = abs(self.posy(j) - meta.posy(j))
103             suma = suma + dmx + dmy
104         return suma
105
106     def descendientes(self):
107         b = self.estado.index(" ")
108         a = self.ascendiente
109         r = list()
110         j = 0
111         while(self.intercambios[b][j] != 9):
112             d = self.estado[:]
113             d[b] = self.estado[self.intercambios[b][j]]
114             d[self.intercambios[b][j]] = " "
115             j = j + 1
116             if( d != a ):
117                 r.append(d)
118         return r
119
120     def printPuzzle(self):
121         print("____")
122         print(' '.join(self.estado[0:3]))
123         print(' '.join(self.estado[3:6]))
124         print(' '.join(self.estado[6:9]))
125         print("_____ h(n)->{ } n->{ } f(n)->{ } explorado->{ }".format(self.h, self.g, self.h
126                                                                           + self.g, self.explorado))
127         print('')
128
```



ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA EN SISTEMAS  
INGENIERÍA EN COMPUTACIÓN



```
130 # INICIO
131 if __name__ == "__main__":
132
133     estados = []
134
135     # OBJETOS LISTA, ESTADO INICIAL Y ESTADO META O FINAL
136     lista = DoubleList()
137     estado_inicial = Puzzle()
138     estado_meta = Puzzle()
139
140     # ESTADO META O FINAL
141     estado_meta.estado = list("12345678 ")
142
143     # ESTADO INICIAL Y PARAMETROS
144     estado_inicial.estado = list(" 74526831")
145     estado_inicial.h = estado_inicial.distanciaManhatan(estado_meta)
146     estado_inicial.g = 0
147     estado_inicial.explorado = 1
148
149     # IMPRIME ENCABEZADOS DEL JUEGO
150     print("Rompecabezas de 8 piezas")
151     print("")
152     print("Estado inicial")
153     estado_inicial.printPuzzle()
154     print("Estado meta ")
155     estado_meta.printPuzzle()
156
157     print("Inicia solucion ")
158
159     # METE A LA LISTA EL NODO INICIAL
160     lista.append(estado_inicial)
161
162     # OBTIENE EL MEJOR NODO DE LA LISTA
163     lista.bestNode()
164     mejorNodo = lista.best.data
165
166     j=0
167     while (j < 2000 and mejorNodo.estado != estado_meta.estado):
168         # OBTIENE DESCENDIENTES DEL MEJOR NODO Y LOS AGREGA A LA LISTA
169         for hijo in mejorNodo.descendientes():
170             elem = Puzzle()
171             elem.estado = hijo
172             elem.h = elem.distanciaManhatan(estado_meta)
173             elem.g = mejorNodo.g + 1
174             elem.explorado = 0
175             elem.ascendiente = mejorNodo.estado
176             lista.append(elem)
177         # OBTIENE EL MEJOR NODO DE LA LISTA
178         lista.bestNode()
179         mejorNodo = lista.best.data
180         j=j+1
181
182     #IMPRIME LISTA COMPLETA CON EL RESULTADO OBTENIDO
183     for v in lista.recover():
184         estados.append(v)
185
186     estados.reverse()
187
188     for e in estados:
189         e.printPuzzle()
```



ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA EN SISTEMAS  
INGENIERÍA EN COMPUTACIÓN



Resultados:

```
In [1]: runfile('C:/Users/ediso/Documents/EPN ESTUDIO/epn 5to semestre/  
Inteligencia Artificial/Laboratorios/Laboratorio 3/Heurística_8_Puzzle_Edison  
Cabrera.py', wdir='C:/Users/ediso/Documents/EPN ESTUDIO/epn 5to semestre/  
Inteligencia Artificial/Laboratorios/Laboratorio 3')  
Rompecabezas de 8 piezas
```

Estado inicial

```
_____  
 7 4  
5 2 6  
8 3 1  
_____  
h(n)->16  n->0  f(n)->16  explorado->1
```

Estado meta

```
_____  
1 2 3  
4 5 6  
7 8  
_____  
h(n)->0  n->0  f(n)->0  explorado->0
```

Inicia solución

```
_____  
 7 4  
5 2 6  
8 3 1  
_____  
h(n)->16  n->0  f(n)->16  explorado->2
```

```
_____  
7  4  
5 2 6  
8 3 1  
_____  
h(n)->15  n->1  f(n)->16  explorado->3
```

```
_____  
7 2 4  
5  6  
8 3 1  
_____  
h(n)->14  n->2  f(n)->16  explorado->5
```

```
_____  
7 2 4  
5 3 6  
8  1  
_____  
h(n)->13  n->3  f(n)->16  explorado->8
```

```
_____  
7 2 4  
5 3 6  
8  1  
_____  
h(n)->12  n->4  f(n)->16  explorado->10
```



ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA EN SISTEMAS  
INGENIERÍA EN COMPUTACIÓN



```
7 2 4
5 3
8 1 6
_____ h(n)->13 n->5 f(n)->18 explorado->15
```

```
7 2 4
5 3
8 1 6
_____ h(n)->12 n->6 f(n)->18 explorado->17
```

```
7 2 4
5 3
8 1 6
_____ h(n)->11 n->7 f(n)->18 explorado->18
```

```
2 4
7 5 3
8 1 6
_____ h(n)->10 n->8 f(n)->18 explorado->19
```

```
2 4
7 5 3
8 1 6
_____ h(n)->11 n->9 f(n)->20 explorado->33
```

```
2 4
7 5 3
8 1 6
_____ h(n)->10 n->10 f(n)->20 explorado->34
```

```
2 4 3
7 5
8 1 6
_____ h(n)->9 n->11 f(n)->20 explorado->35
```

```
2 4 3
7 5
8 1 6
_____ h(n)->10 n->12 f(n)->22 explorado->134
```



ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA EN SISTEMAS  
INGENIERÍA EN COMPUTACIÓN



```
2 4 3
7 1 5
8 6
_____ h(n)->9 n->13 f(n)->22 explorado->137
```

```
2 4 3
7 1 5
8 6
_____ h(n)->8 n->14 f(n)->22 explorado->138
```

```
2 4 3
1 5
7 8 6
_____ h(n)->7 n->15 f(n)->22 explorado->139
```

```
2 4 3
1 5
7 8 6
_____ h(n)->6 n->16 f(n)->22 explorado->140
```

```
2 3
1 4 5
7 8 6
_____ h(n)->5 n->17 f(n)->22 explorado->141
```

```
2 3
1 4 5
7 8 6
_____ h(n)->4 n->18 f(n)->22 explorado->142
```

```
1 2 3
4 5
7 8 6
_____ h(n)->3 n->19 f(n)->22 explorado->143
```

```
1 2 3
4 5
7 8 6
_____ h(n)->2 n->20 f(n)->22 explorado->144
```

```
1 2 3
4 5
7 8 6
_____ h(n)->1 n->21 f(n)->22 explorado->145
```

```
1 2 3
4 5 6
7 8
_____ h(n)->0 n->22 f(n)->22 explorado->146
```