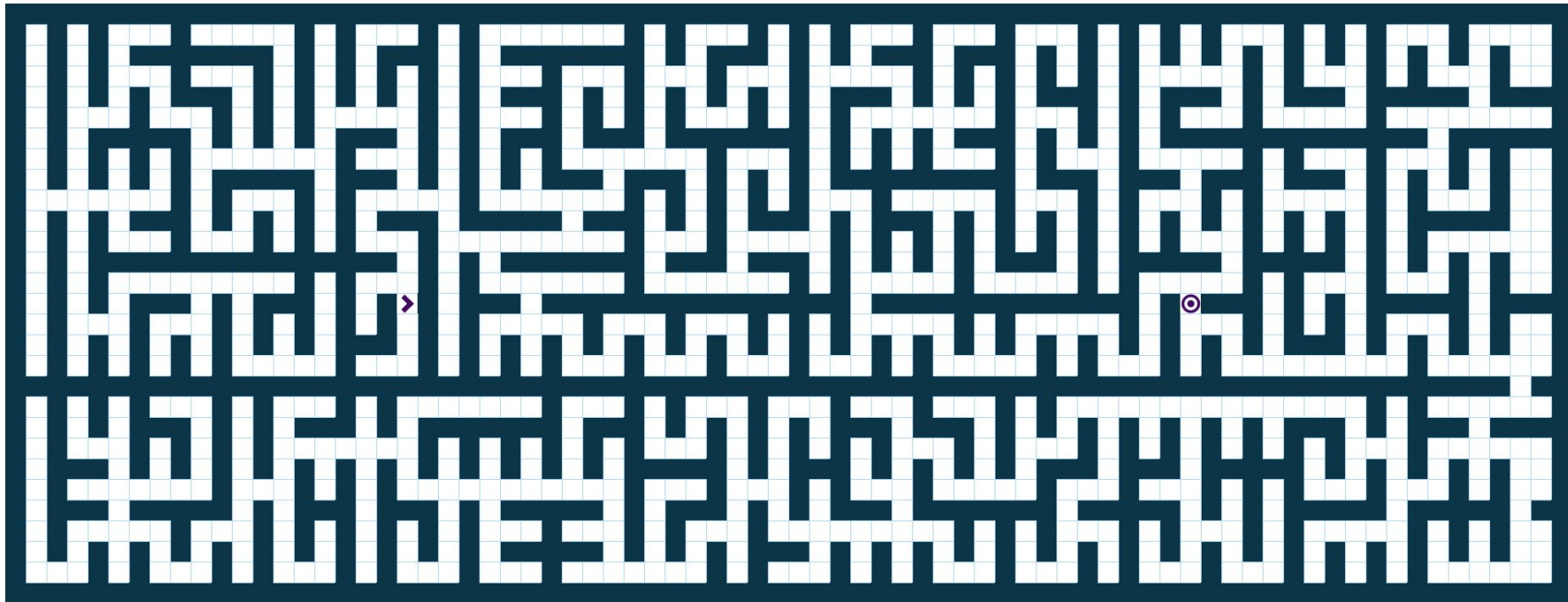

Ayudantía 1

Pablo Flores : ptflores1@uc.cl
Sergio M Appel : matamalaappels@uc.cl

Búsqueda

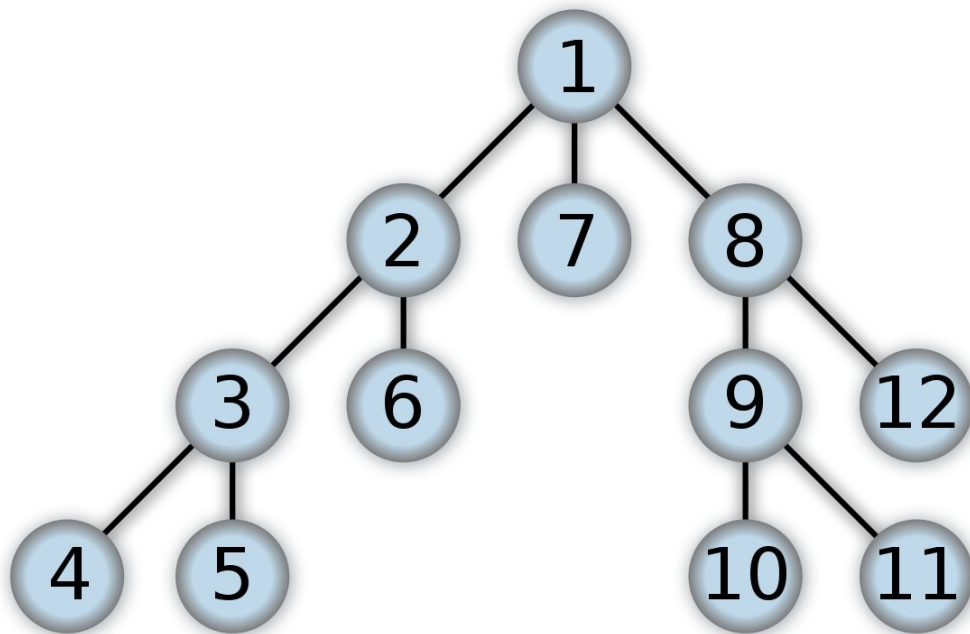


Algoritmos

- DFS
- BFS
- Dijkstra
- A*

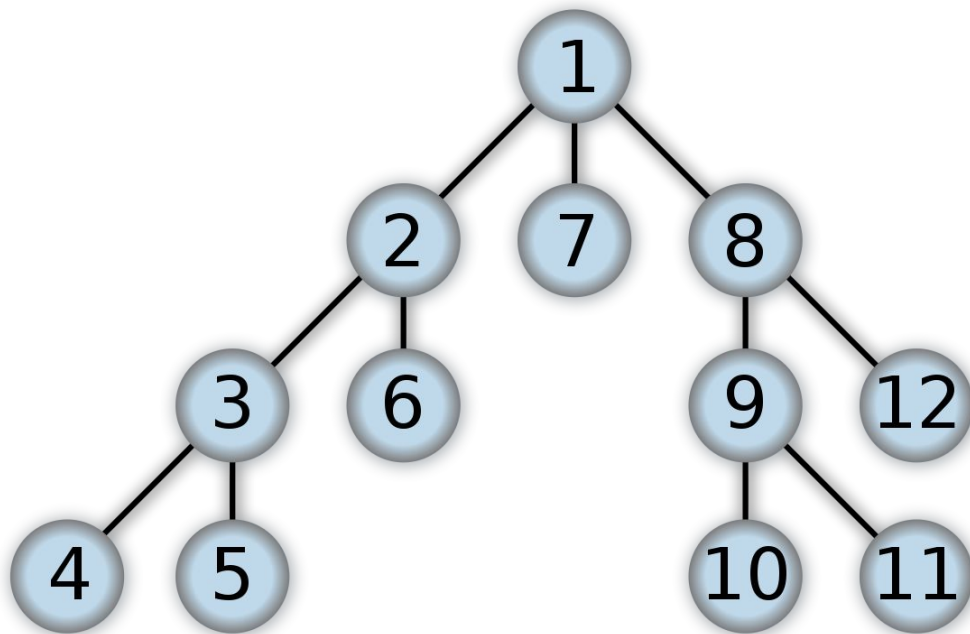
DFS

Búsqueda en profundidad. Tarjet -> 6



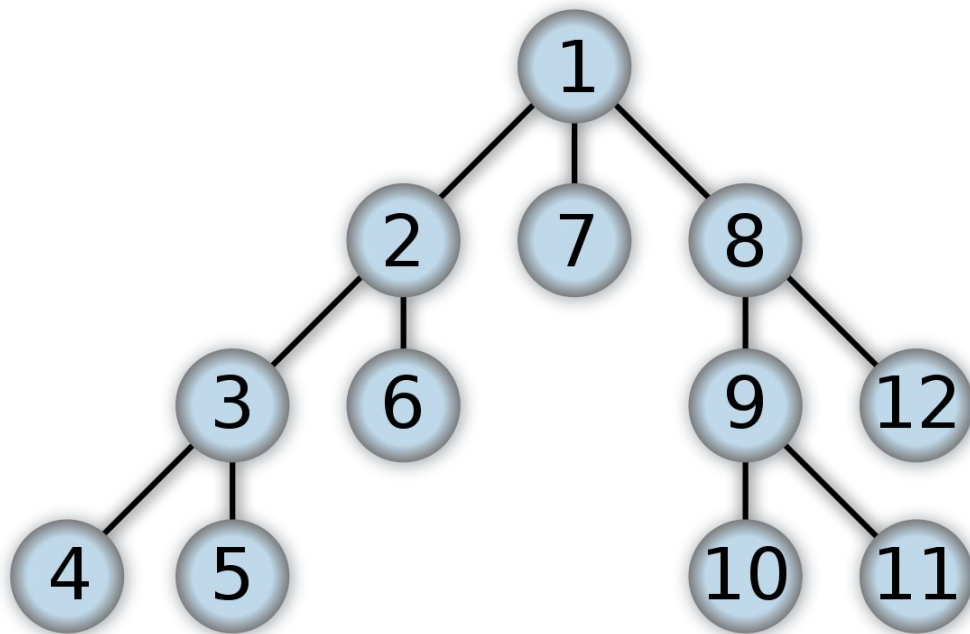
BFS

Búsqueda en amplitud. Tarjet -> 6



BFS

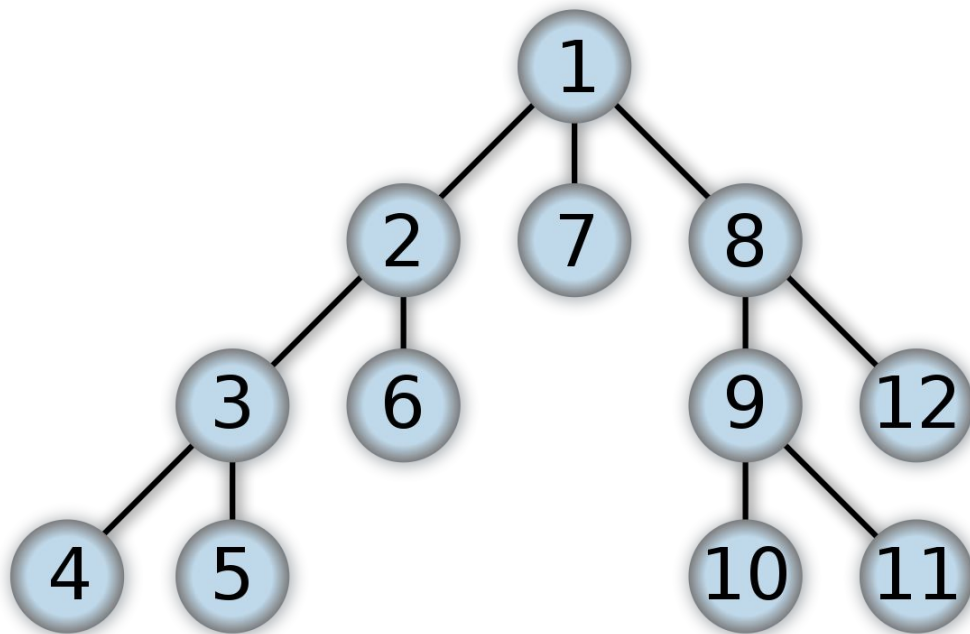
Búsqueda en amplitud. Tarjet -> 6



Cola: 8 - 7 - 2

BFS

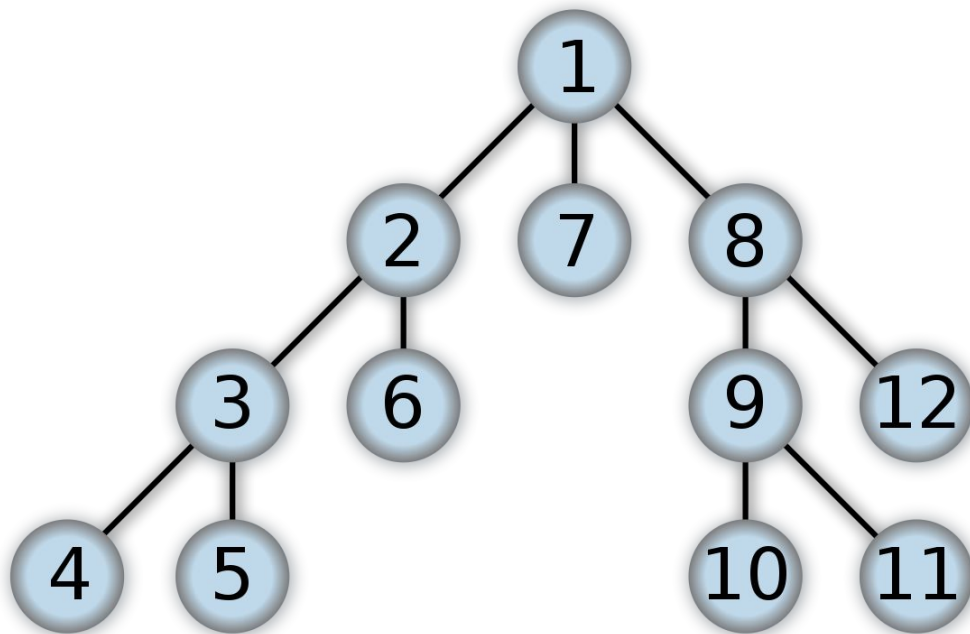
Búsqueda en amplitud. Tarjet -> 6



Cola: 7 - 2 - 12 - 9

BFS

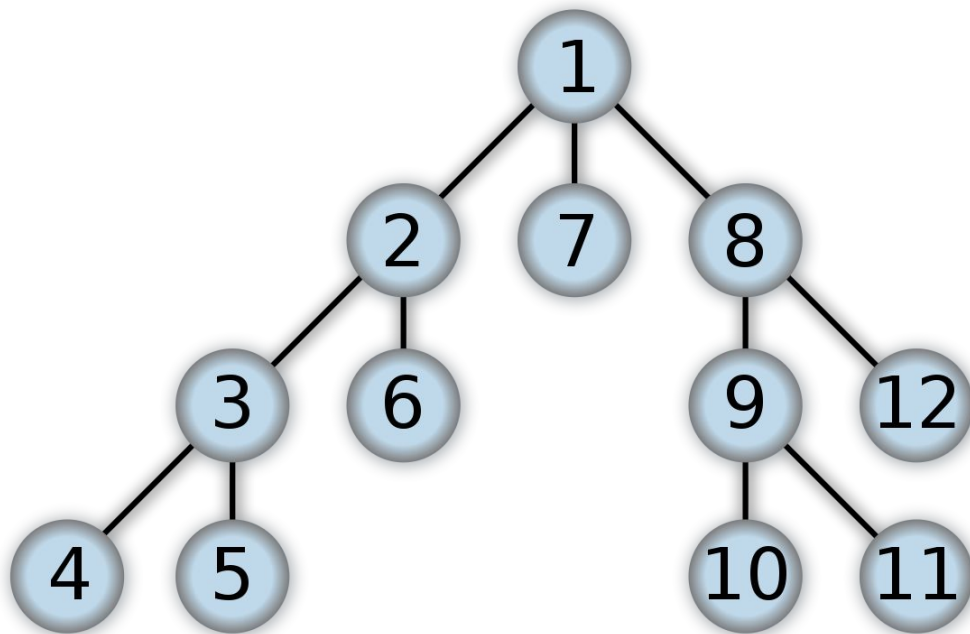
Búsqueda en amplitud. Tarjet -> 6



Cola: 2 - 12 - 9

BFS

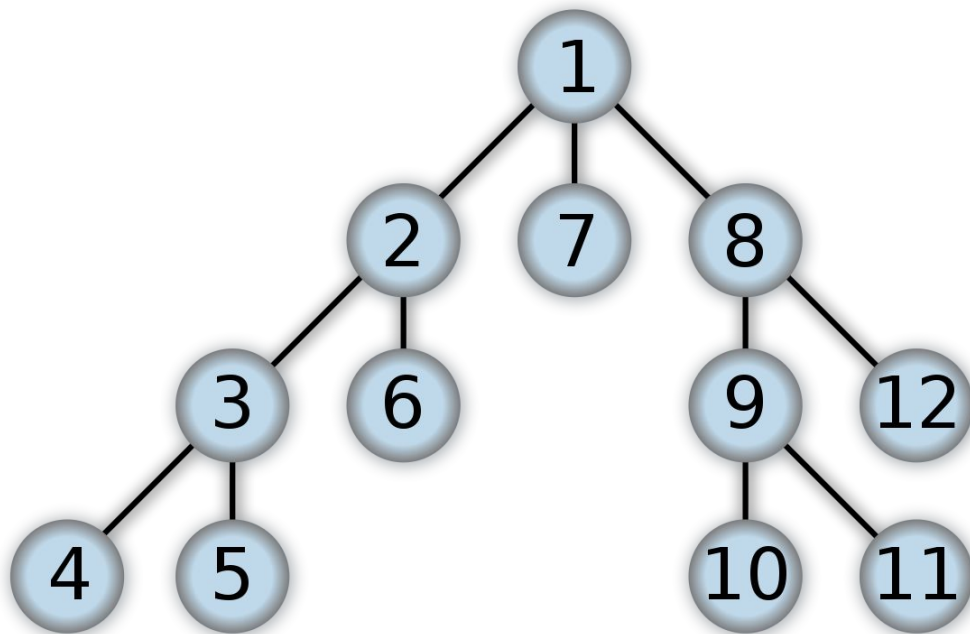
Búsqueda en amplitud. Tarjet -> 6



Cola: 12 - 9 - 6 - 3

BFS

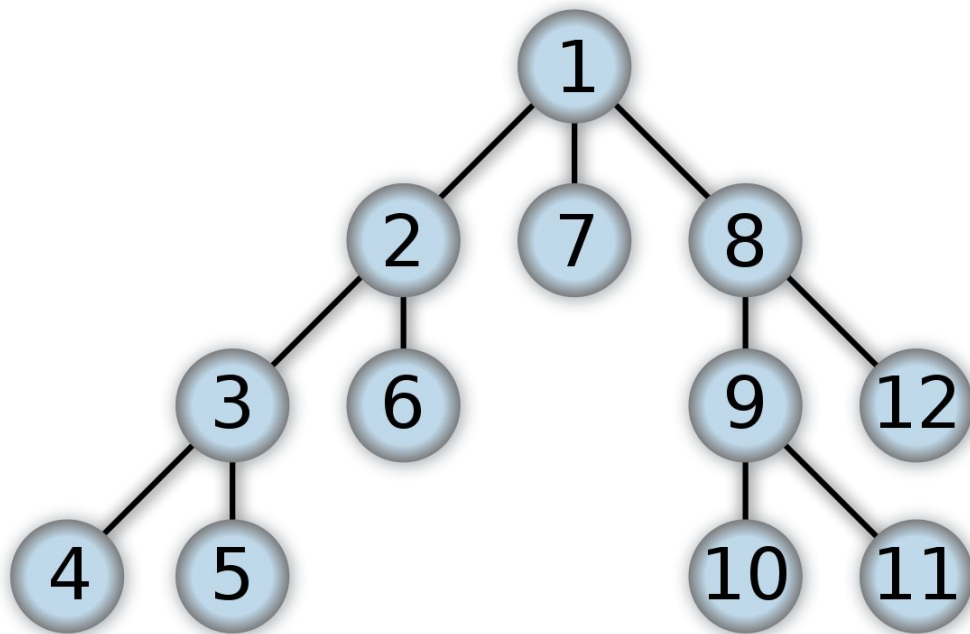
Búsqueda en amplitud. Tarjet -> 6



Cola: 9 - 6 - 3

BFS

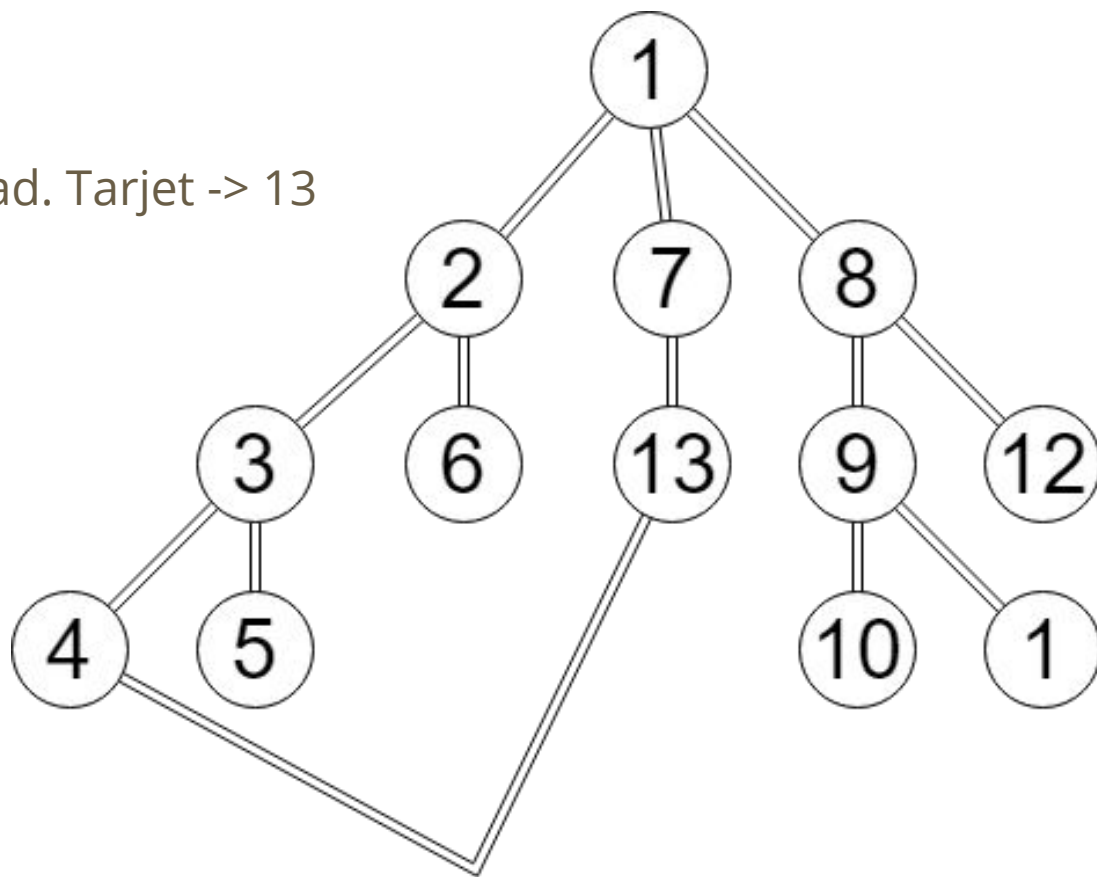
Búsqueda en amplitud. Tarjet -> 6



Cola: 6 - 3 - 11 - 10

DFS

Búsqueda en profundidad. Tarjet -> 13



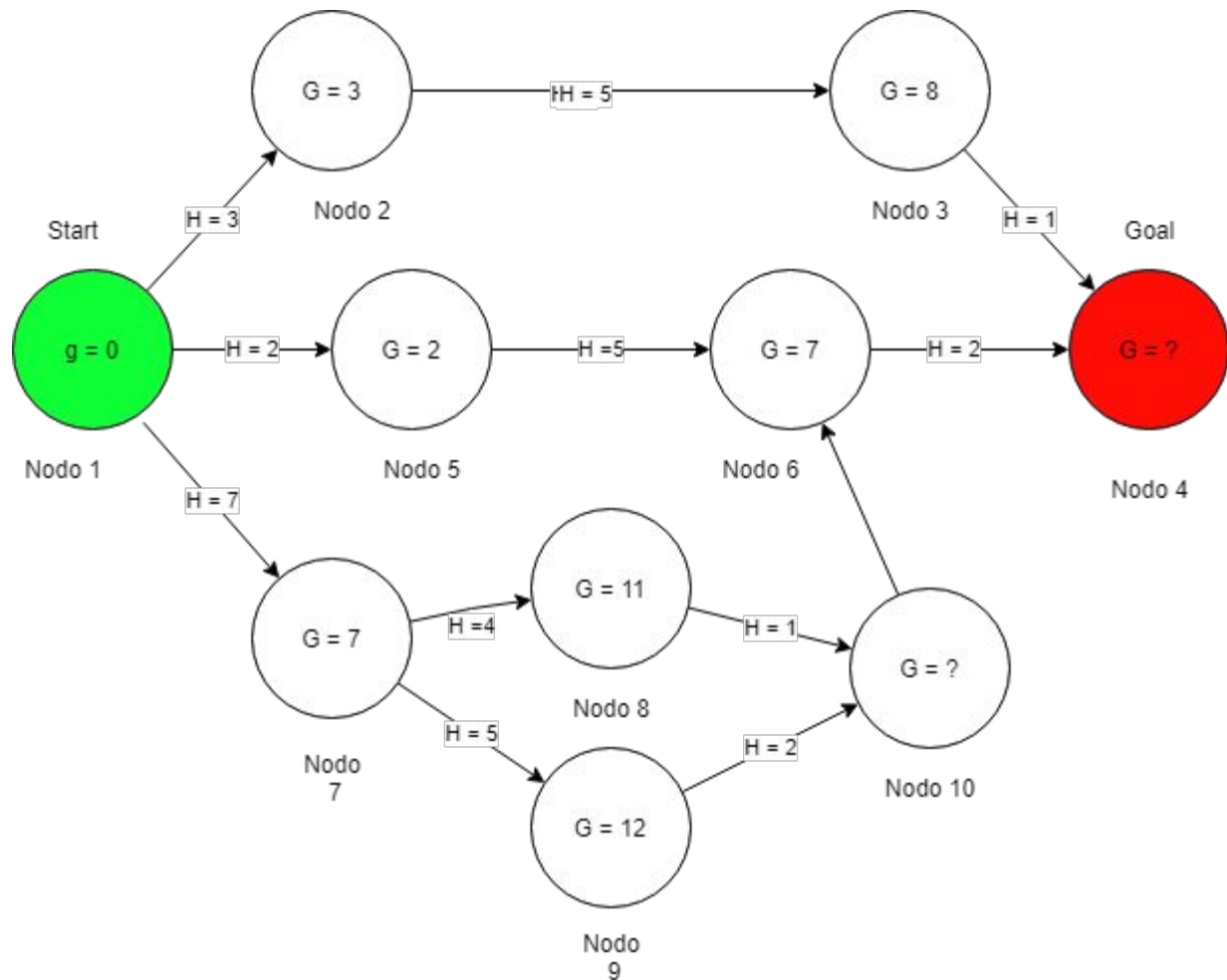
Dijkstra

Ejemplo en whiteboard

A*

$$f(n) = g(n) + h(n)$$

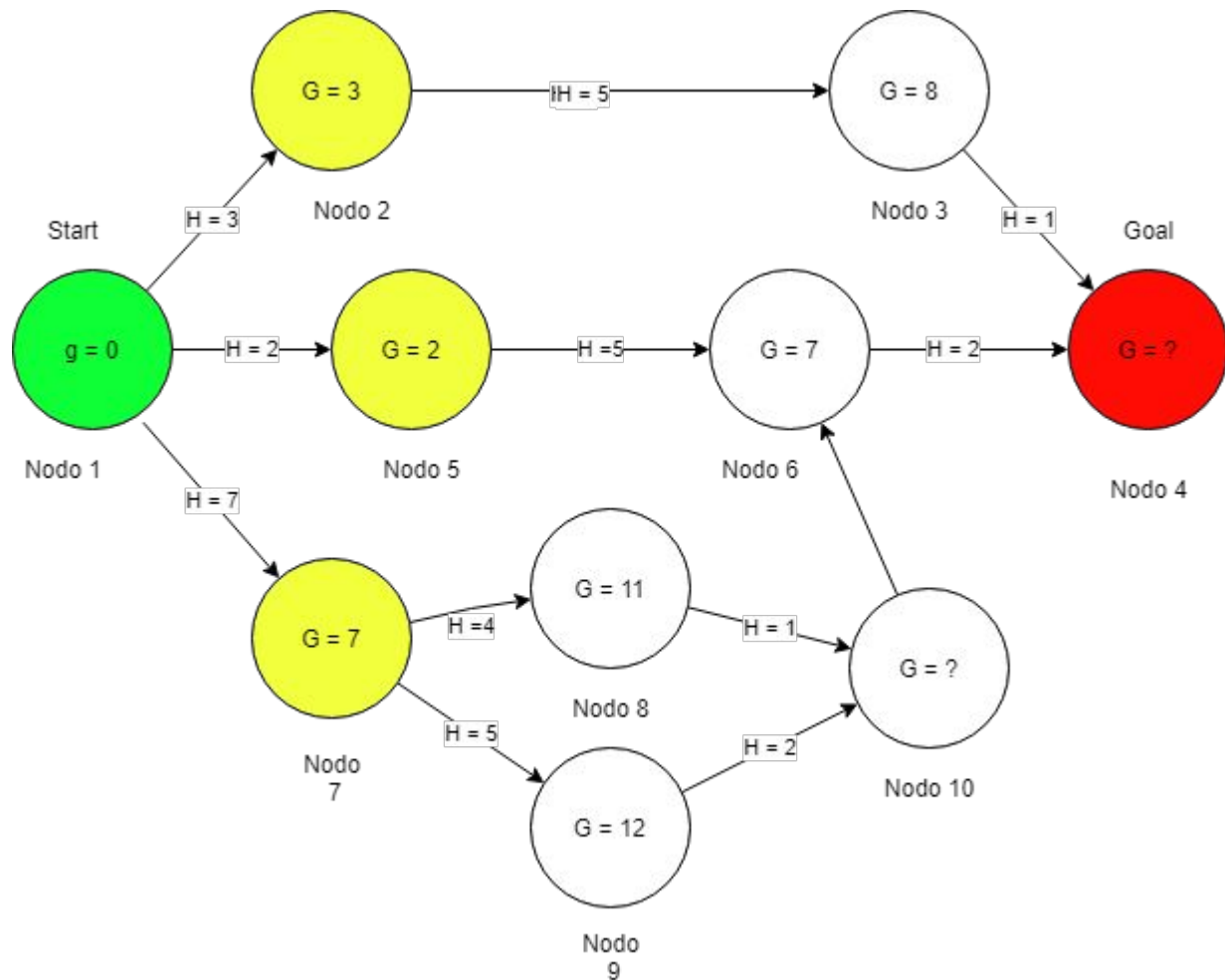
Open = {}



A*

$$f(n) = g(n) + h(n)$$

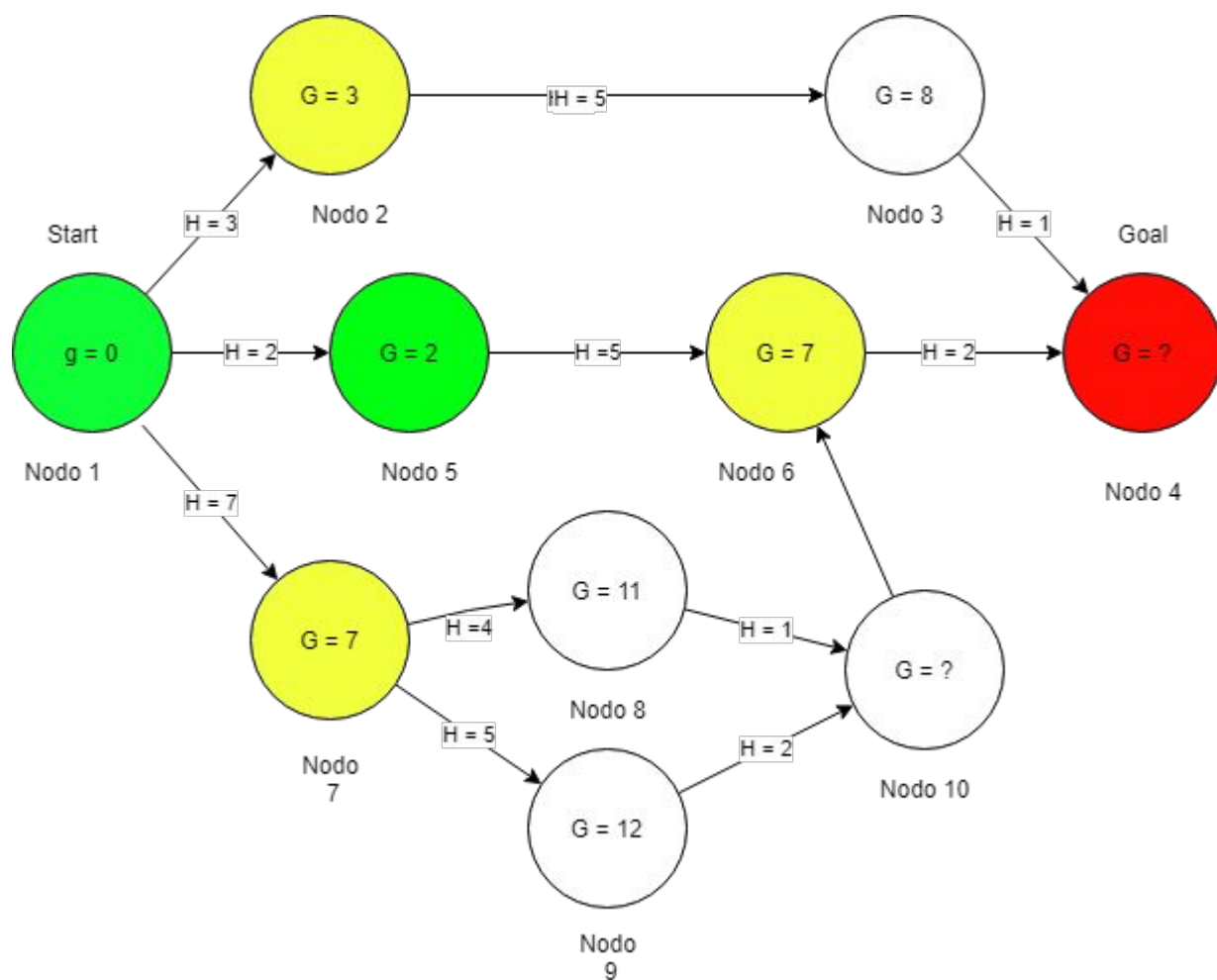
Open = {N5 - N2 - N7}



A*

$$f(n) = g(n) + h(n)$$

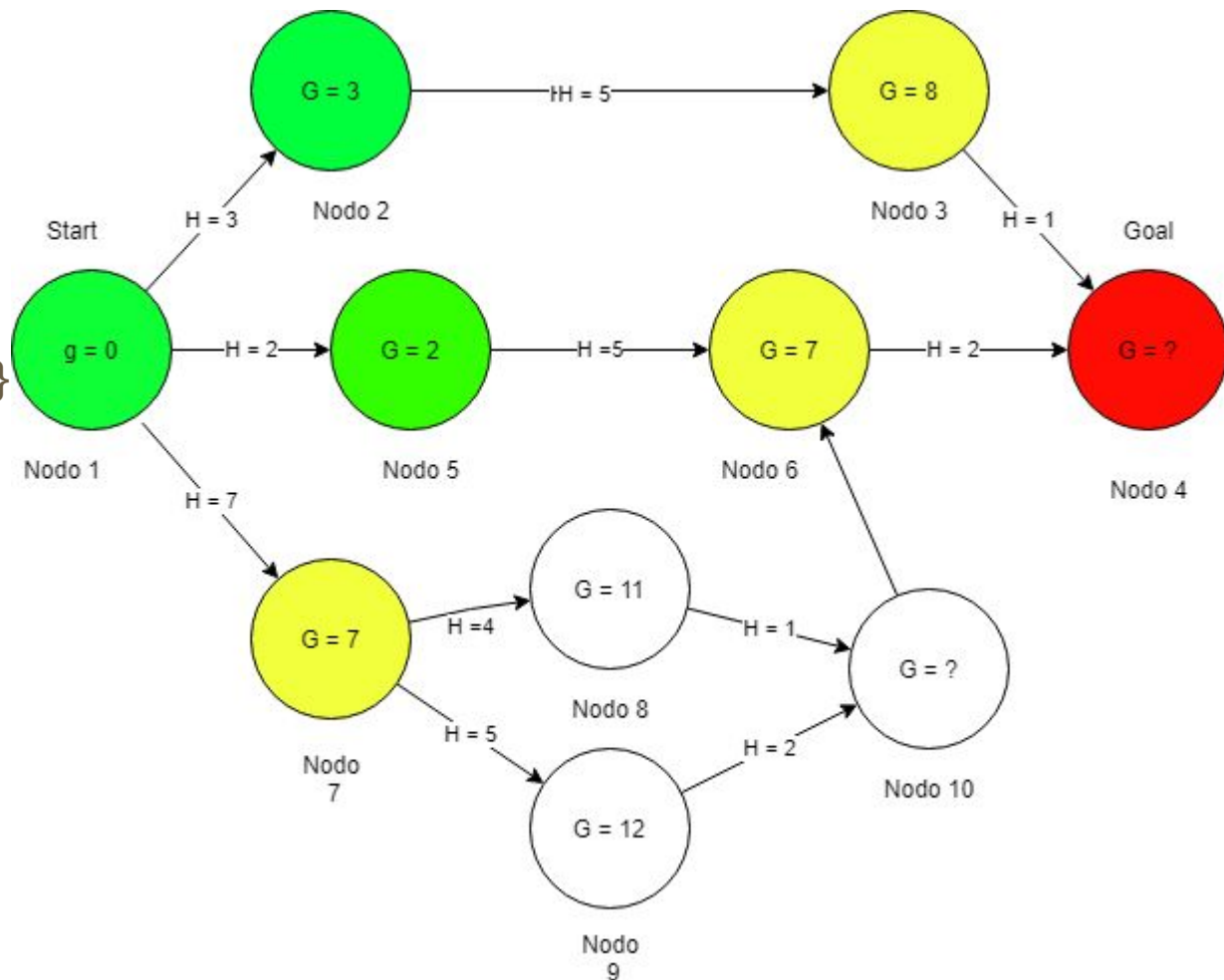
Open = {N2 - N6 - N7}



A*

$$f(n) = g(n) + h(n)$$

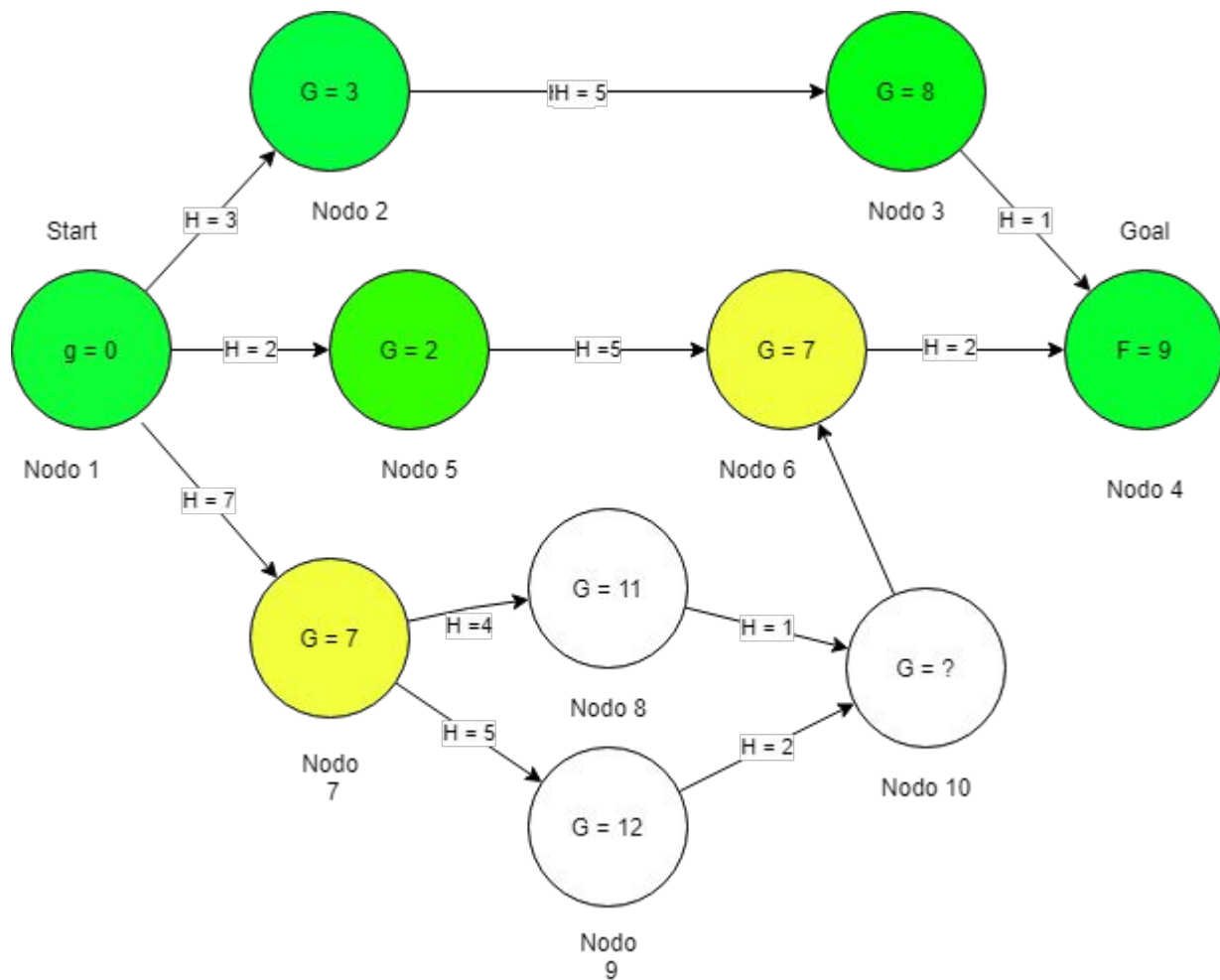
Open = {N3 - N6 - N7}



A*

$$f(n) = g(n) + h(n)$$

Open = { N6 - N7 }

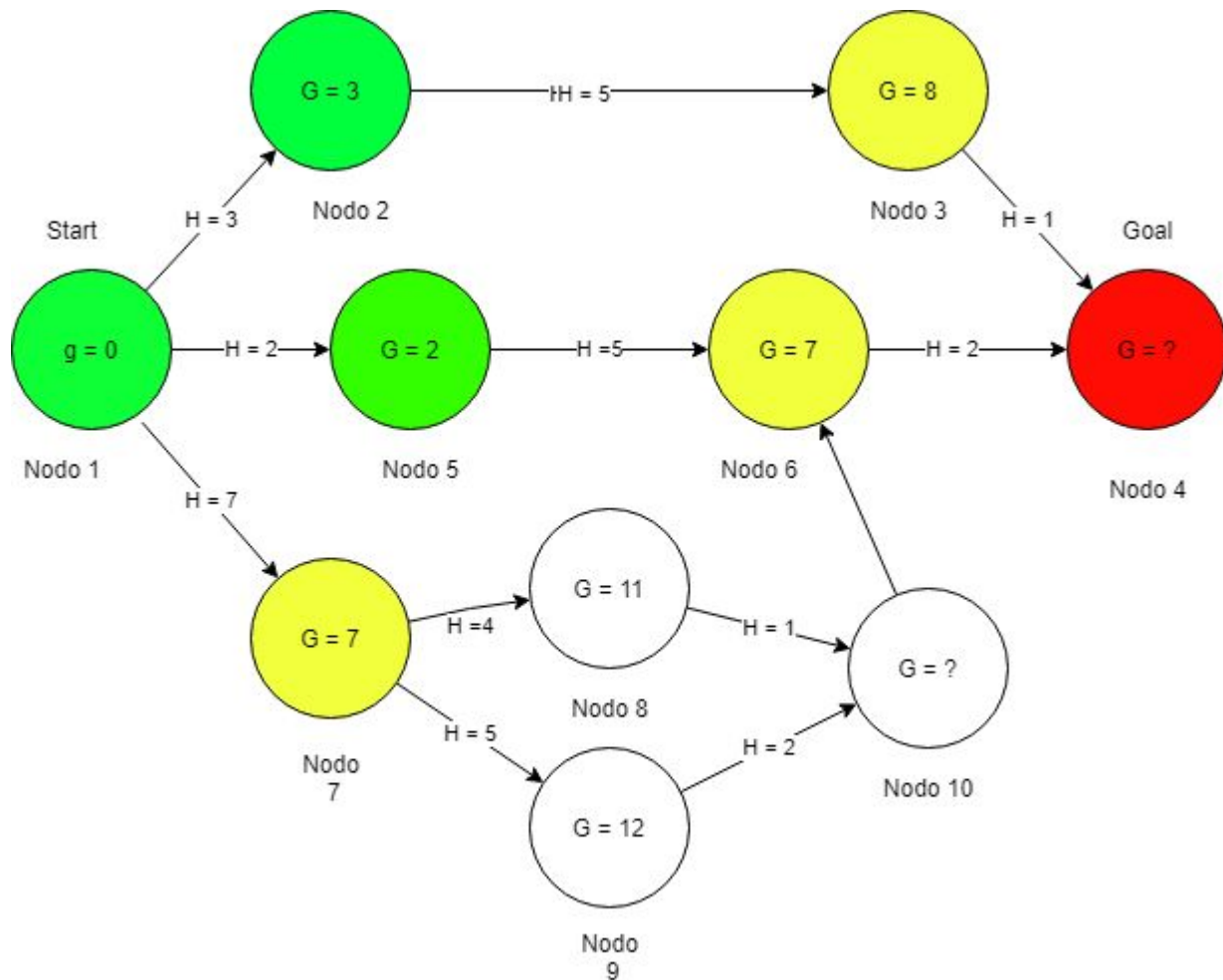


A*

Des - Empate en la open

$f(n_1) = f(n_2)$

Open = {? ? ?}



A*

Des - Empate en la open

Open = {? ? ?}

#prob	#exp	#gen	sol	tiempo	maxsubopt
1	163769	312719	45	23.33	1.00
2	210855	391609	42	28.52	1.00
3	240337	461748	42	26.08	1.00
4	199299	376533	41	21.66	1.00
5	677552	1264458	46	73.34	1.00

#prob	#exp	#gen	sol	tiempo	maxsubopt
1	151817	290844	45	15.44	1.00
2	170564	320461	42	18.23	1.00
3	198327	384013	42	19.52	1.00
4	191259	362370	41	18.73	1.00
5	620168	1162970	46	60.47	1.00

A*

$$f(n) = g(n) + h(n)$$

```
def fvalue(self, g, h):  
    #implementamos la comparacion de forma i  
    # aca ponderamos el costo total por un n  
    # para que efectivamente tengan priorida  
    # en nuestro caso como las sol son todas  
    #return g + h  
    return 8000*(g + h * self.weigth) - h
```

Weigthed A*

Spoilers

W = 5

#prob	#exp	#gen	sol	tiempo	maxsubopt
1	2130	4357	91	0.30	2.60
2	6706	11896	76	0.78	2.38
3	4851	9247	68	0.73	2.12
4	5756	10030	71	0.73	2.29
5	1883	3140	74	0.28	2.06
6	4703	8454	85	0.53	2.74

W = 1.5

#prob	#exp	#gen	sol	tiempo	maxsubopt
1	10201	20164	45	1.77	1.15
2	24813	47294	42	3.73	1.17
3	29774	58225	42	4.47	1.17
4	22263	43567	43	2.89	1.23
5	7087	14053	46	0.97	1.21

Heurística

- Admisibilidad
- Consistencia

Initial State

1	2	3
8		4
7	6	5

Goal State

2	8	1
	4	3
7	6	5

Heurística - Admisibilidad

h admisible SSI :

- $h(s) \leq h^*(s)$ for all s
 - Una heurística nunca sobrestima el costo para llegar al estado objetivo.

Heurística - consistencia

h es consistente SSI :

- $h(s) = 0$ for all s subconjunto de G
- $h(s) \leq c(s, s') + h(s')$ para todo vecino s' de s
 - $s = \text{Padre}, s' = \text{Hijo}$