

5.1 Algoritmo A*

Victor Misael Escalante Alvarado, A01741176

Parte 1

Captura del codigo funcinando :



Parte 2

Captura del código funcionando

```
> OUTLINE
> TIMELINE
PROBLEMS  JUPYTER
---
1 3 5
4 2
7 8 6
---
1 3 5
4 2
7 8 6
---
1 3 5
4 2
7 8 6
---
1 3
4 2 5
7 8 6
---
1 2 3
4 5
7 8 6
---
1 2 3
4 5 6
7 8 _
---
1 2 3
4 5 6
7 8 _
---
1 2 3
4 5 6
7 8 _

while current in came_from:
    path.append(current)
    current = came_from[current]
path.append(inicio)
return path[::-1] # Invertimos el camino

# Generar vecinos
for vecino in obtener_vecinos(current):
    tentative_g_score = g_score[current] + 1
    if vecino not in g_score or tentative_g_score < g_score[vecino]:
        came_from[vecino] = current
        g_score[vecino] = tentative_g_score
        f_score[vecino] = g_score[vecino] + manhattan_dist(vecino, objetivo)
        heapq.heappush(open_set, (f_score[vecino], vecino))

return [] # Si no se encuentra solución

# Caso de prueba
inicio = ('3', '2', '_', '7', '1', '4', '6', '5', '8')
objetivo = ('1', '2', '3', '4', '5', '6', '7', '8', '_')

print("Pasos: ", len(aStar_puzzle(inicio, objetivo)))
print("Camino:")
camino = aStar_puzzle(inicio, objetivo)
for paso in camino:
    print("\n".join(" ".join(paso[i:i+3]) for i in range(0, len(paso), 3)))
    print("----")

Paos: 25
Camino:
3 2 _
7 1 4
```

Enlace a colab:

[https://drive.google.com/file/d/1c-Ebth_Zj5cEDx-QTwoL50emdsRyXUpo/view?usp=sharing]