

Ingeniería Mecatrónica

Inteligencia Artificial

Profesor: Mauricio Alejandro Cabrera Arellano

6° Semestre

Practica 2.- Djikstra

Autor: Paulo Salvatore Partida Escamilla

Matricula: 19310483

Contenido

[1. Link repositorio 3](#_Toc99927789)

[1. Grafos missing. **¡Error! Marcador no definido.**](#_Toc99927790)

1. Link repositorio

Que tal profesor!

Le comparto el link de mi repositorio de Github y la prueba de que ya lo añadí como colaborador:

<https://github.com/PauloCeti/InteligenciaArtificial>D

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

1. Primer boceto de la algoritmia y el entorno

Okay, comenzaré aclarando que es el primer boceto que tengo de la algoritmia a seguir para solucionar el problema, la cual es posible que se vea modificada, alterada o corregida durante la programación de este.

“Camino más fácil para conquistar a tu crush para gente como Paulo”

* Inicio. Tienes identificada a tu crush
* 1. Presentarte sin hablarle antes
* 2. Buscarla en redes sociales y presentarte por ahí
* 3. Contactar a un amigo de ella y pedirle que los presente
* 4. Invitarla a salir
* 5. Decirle que te gusta
* 6. Investigar/ preguntarle sobre sus gustos
* 7. Establecer platica con ella sobre lo que les gusta a los dos
* 8. Darle un regalo
* 9. Darle pequeños detalles
* 10. Conocer a sus amigos y hacerlos mis amigos
* 11. Besarla
* 12. Demostrar atención por su bienestar (físico y mental)
* Final. Conquistas a tu crush

Al ser un grafo muy personal, la distribución de dificultad entre cada uno de los nodos (que sería el peso de las aristas, la “dificultad”) es completamente subjetiva y aún no está 100% definida, planea aterrizarse una vez visto el grafo, para poder definir las lineas de avance (la intención es que la escala vaya del 1-10) y que el camino más corto sea el siguiente: **2-6-7-9-12-4-11.**

Se tiene un primer boceto de la comunicación entre nodos, sin embargo, puede variar y aún está por determinarse si es dirigido o si las aristas son unidireccionales:

|  |  |
| --- | --- |
| Nodo | Posibles nodos aledaños |
| 1 | 4,5,6,8 |
| 2 | 4,5,6,9 |
| 3 | 4,5,6,10 |
| 4 | 8,9,11,12 |
| 5 | 11, Fin |
| 6 | 7,8,9 |
| 7 | 4,5,8,9,10,12 |
| 8 | 4,5 |
| 9 | 4 |
| 10 | 6 |
| 11 | 4,5,12, Fin |
| 12 | 4,5,11, Fin |

Respecto al funcionamiento del programa (como primer boceto), se implementarían diccionarios, en donde cada entrada es un nodo y entre las propiedades de la entrada, se encuentren tuples o listas o quizá otro diccionario, en donde se pueda extraer el nodo al que se apunta y que peso tiene, para que se pueda extraer el peso más pequeño y abstraer el nodo con ese camino más pequeño, de ese modo, el bucle analizaría la entrada del diccionario con el mismo ID que el nodo y se avanzaría así hasta encontrar el fin.