**Laboratorio S14. AYED-02**

Jefer Alexis González Romero

**DOCUMENTO TÉCNICO**

**Implementación del algoritmo de recorrido BFS para la representación de matriz de adyacencias**

**Requisitos**

**Especificación**

grafo =

**1**

**2**

**5**

**4**

**7**

**6**

**3**

grafo.bfs(3)

**1**

**2**

**5**

**4**

**7**

**6**

**3**

3 4 5 6 7

**Entrada:**

Se tiene un grafo y se recibe un vértice.

**Salida:**

Los vértices que son alcanzables desde el vértice dado.

**Diseño**

**Estrategia**

Se agregará un nuevo atributo “data\_props” donde estará la información de cada vértice, como el color (para saber si ya se pasó por ahí), la distancia a la que está de un vértice dado, un phi que es el vértice que está antes de él, y por último la ruta que se tuvo que hacer para llegar hasta ahí.

Se hace la función “get\_neighbors” que dará los vecinos de un vértice dado, ya por último la función bfs que buscará los vértices alcanzables por uno dado, para esto para cada vértice se le asignarán unas propiedades iniciales en” data\_props”, a excepción del que nos dan todos tendrán como color “WHITE” y distancia infinita representando que no se puede alcanzar inicialmente. Del vértice que partimos comenzará con color “GRAY” y distancia 0. Ahora en una cola se van poniendo los que se alcanzan y se quitan cuando ya se agregaron todos sus vecinos (solo se agregan si aún no se ha pasado por ahí), cuando pasa esto se colocan en “BLACK”. Si aun no se han tomados los vecinos de un vértice este tendrá color “GRAY”. Este proceso se termina cuando en la cola no queda ningún elemento.

Adjunto el programa con nombre “BFS Matriz de adyacencia”.

**Casos prueba**

Para los casos prueba se va a usar el siguiente árbol:

grafo =

**1**

**2**

**5**

**4**

**7**

**6**

**3**

|  |  |  |
| --- | --- | --- |
| Entrada | Justificación | Salida |
| grafo.bfs(6) | Vértice que no alcanza ningún otro aparte de él. | 6 |
| grafo.bfs(1) | Vértice con alcance de todos | 1 2 3 4 5 6 7 |
| grafo.bfs(5) | Vértice con solo alcance de sus vecinos | 4 5 6 7 |

**Análisis**

**Temporal**

En el mejor de los casos T(n) = Ω(1)

En el peor de los casos T(n) = O(n)

**Código**

**Documentación**

Dentro del código

**Prototipo**

Dentro del programa con nombre “BFS Matriz de adyacencia”.

La gráfica que ilustra el grafo usado en el prototipo es el siguiente:

**1**

**2**

**5**

**4**

**7**

**6**

**3**

**Casos de prueba**

1. Vive en la ciudad 3

**1**

**2**

**5**

**4**

**7**

**6**

**3**

1. Vive en la ciudad 5

**1**

**2**

**5**

**4**

**7**

**6**

**3**

1. Vive en la ciudad 6

**1**

**2**

**5**

**4**

**7**

**6**

**3**

1. Vive en la ciudad 1

**1**

**2**

**5**

**4**

**7**

**6**

**3**