



Universidad Autónoma de Nuevo León
Facultad de Ciencias Físico-Matemáticas



Tarea 3

Alumnos:

Leonardo Martin Vazquez Cruz 1679807

Bryan Aguirre Tudon

Alexis Emiliano Eguia Sifuentes

Materia: Inteligencia Artificial

Docente: Juan Pablo Rosas Baldazo

Programar un DFS y un BFS considerando lo siguiente:

Deben de tener al menos dos instancias para probar sus algoritmos (De preferencia que sean las mismas dos para ambos algoritmos). Para ello, las pueden hacer a mano o utilizar el generador de instancias de la tarea anterior. Cada instancia debe de ser, al igual que la tarea anterior, de mínimo 20 nodos.

Comparar y graficar los resultados. Pueden comparar el tiempo que tarda cada algoritmo en encontrar un nodo determinado. También, pueden revisar si la forma del grafo afecta el desempeño de los algoritmos, es decir, si el grafo es completo o no, si es dirigido o no, la cantidad de nodos, etc.

Lo que si pueden agregar es pseudocódigo o diagramas de flujo de sus algoritmos, tablas de resultados (con datos relevantes), descripción de sus experimentos, descripción de sus resultados y conclusiones.

Resultados

DFS

```
PS C:\Users\52813\Downloads> c:; cd 'c:\Users\52813\Downloads'; & 'C:\Users\52813\AppData\Local\Programs\Python\Python311\python.exe' 'c:\Users\52813\.vscode\extensions\ms-python.python-2022.18.2\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '52212' '--' 'c:\Users\52813\Downloads\DFS.py'
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Tiempo de ejecución:
0.07199335098266602
```

BFS

```
PS C:\Users\52813\Downloads> c:; cd 'c:\Users\52813\Downloads'; & 'C:\Users\52813\AppData\Local\Programs\Python\Python311\python.exe' 'c:\Users\52813\.vscode\extensions\ms-python.python-2022.18.2\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '52135' '--' 'c:\Users\52813\Downloads\BFS (1).py'
0 1 2 7 8 3 6 9 12 4 5 10 11 13 14 15 16 17 18
Tiempo de ejecución:
1.0159976482391357
```

Implementación

En BFS, necesitamos mantener una estructura de datos separada para rastrear los nodos de árbol/graph que aún no se han visitado. Esto se hace fácilmente iterativamente usando la estructura de datos de la queue.

A diferencia de BFS, DFS no necesita ninguna estructura de datos adicional para almacenar los nodos de árbol/graph.

Memoria de los algoritmos

La memoria es tomada por DFS/BFS depende en gran medida de la estructura de nuestro árbol/gráfico. La memoria máxima que ocupa DFS (es decir, la pila de llamadas) es igual a la profundidad del árbol, y la memoria máxima que ocupa BFS es igual al ancho del árbol.

Tiempo de Ejecucion

Se comparan los algoritmos (DFS Y BFS) y nos damos cuenta que el algoritmo BFS es mas constante que el algoritmo DFS en cuanto a los tiempos de ejecucion o el tiempo en que tardan en dar los resultados del algoritmo.

Para medir cada uno de los tiempos de ejecucion se probaron con diferentes cantidades de nodos.

