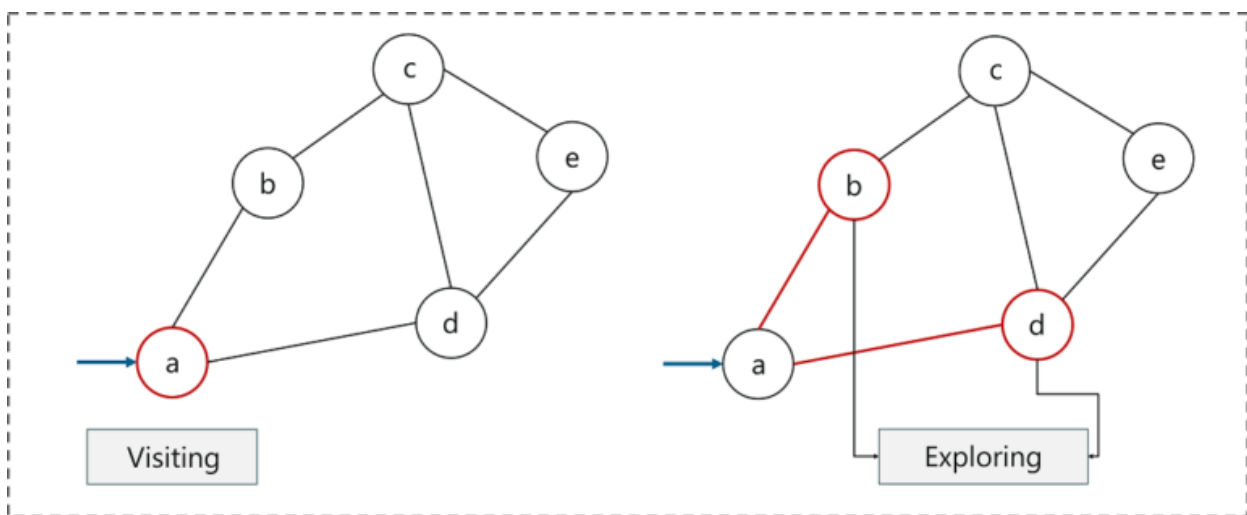


BREADTH-FIRST SEARCH

Graph traversal adalah metodologi yang umum digunakan untuk menentukan posisi titik dalam graph. Sebelumnya, graph adalah kumpulan dari titik (node) dan garis dimana pasangan-pasangan titik (node) tersebut dihubungkan oleh segmen garis. Node ini biasa disebut simpul (vertex) dan segmen garis disebut sisi/ruas (edge).

Graph traversal berarti mengunjungi setiap simpul (vertex) dan tepi (edge) tepat satu kali dalam urutan yang jelas atau sistematis dari sebuah grafik. Saat menggunakan algoritma graph traversal tertentu, pastikan terlebih dahulu setiap simpul dari graph dikunjungi tepat satu kali. Urutan di mana simpul dikunjungi sangat penting dan bergantung pada algoritma yang ingin dipecahkan.



Sumber: edureka.co

1. Mengunjungi sebuah node (visiting a node): seperti namanya, mengunjungi sebuah node berarti mengunjungi atau memilih sebuah node
2. Menjelajahi node (exploring a node): menjelajahi node yang berdekatan (node tetangga) dari node yang dipilih

Terdapat dua algoritma pada graph traversal yakni Depth-First Search (DFS) dan Breadth-First Search (BFS).

Pengertian Breadth-First Search

Breadth-first search (BFS) atau Breadth-first traversal adalah algoritma traversing yang digunakan untuk melintasi atau mencari semua simpul atau node dari suatu struktur data tree atau graph.

Pada algoritma BFS, pencarian dimulai dari pemilihan node awal kemudian dilanjutkan dengan pencarian bertahap level demi level, memeriksa seluruh node pada kedalaman tertentu sebelum masuk ke level yang lebih dalam lagi hingga ditemukan tujuan atau goal state-nya.

Cara kerja algoritma BFS

Queue adalah struktur data abstrak yang mengikuti prinsip First-In-First-Out (FIFO) yang mana data yang dimasukkan terlebih dahulu akan diakses lebih dulu. Struktur data ini terbuka di kedua ujungnya, di mana satu ujung selalu digunakan untuk memasukkan data (enqueue) dan ujung lainnya digunakan untuk menghapus data (dequeue).

Sekarang mari kita lihat langkah-langkah yang terlibat dalam melintasi grafik dengan menggunakan algoritma breadth-first search:

Langkah 1: Ambil antrian kosong.

Langkah 2: Pilih node awal (mengunjungi node) dan masukkan ke dalam antrian.

Langkah 3: Asalkan antrian tidak kosong, ekstrak node dari antrian dan masukkan node anaknya (menjelajahi node) ke dalam antrian.

Langkah 4: Cetak simpul yang diekstraksi

Pseudo Code

```
Input: s as the source node

BFS(G, s)
let Q be queue.
Q.enqueue(s)

mark s as visited
while (Q is not empty)
v = Q.dequeue()

for all neighbour w of v in Graph G
if w is not visited
Q.enqueue(w)
mark w as visited
```


Gambar di atas menggambarkan proses end-to-end dari Breadth-First Search Algorithm. Berikut penjelasan dari gambar diatas:

1. Simpul atau node ‘a’ akan dianggap sebagai simpul akar dan masukkan ke dalam antrian.
2. Kemudian ekstrak node 'a' dari antrian dan masukkan node anak 'a', yaitu, 'b' dan 'c'.
3. Cetak simpul 'a'.
4. Dapat dilihat antrian tidak kosong dan memiliki simpul 'b' dan 'c'. Karena 'b' adalah simpul pertama dalam antrian, maka diekstrak dan kemudian masukkan simpul anak dari 'b', yaitu, simpul 'd' dan 'e'.
5. Ulangi langkah ini sampai antrian kosong. Perhatikan bahwa node yang sudah dikunjungi tidak boleh ditambahkan ke antrian lagi.

Kelebihan algoritma BFS

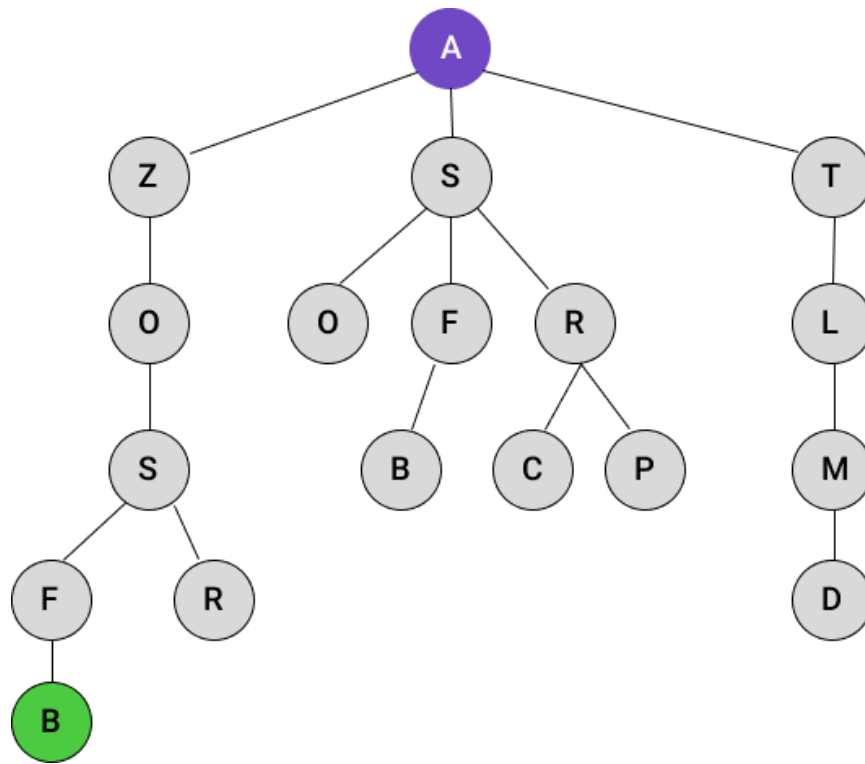
- Tidak akan menemui jalan buntu (complete)
- Jika algoritma menemui beberapa solusi, BFS dapat menemukan solusi yang paling minimum optimal

Kekurangan algoritma BFS

- Membutuhkan memori yang banyak, karena menyimpan keseluruhan node dalam satu pohon / graph
- Membutuhkan waktu yang lebih lama, karena menguji seluruh node pada graph di level ke-n

Studi Kasus

Node	Queue
Inisiasi	A
Level-1	Z S T
Level-2	O O F R L
Level-3	S B(Goals) C P M



Pada studi kasus kali ini, tugas algoritma BFS adalah menemukan jarak terdekat dari node A ke B. Tugas yang dilakukan oleh algoritma BFS adalah mengecek node pada setiap level apakah memiliki node yang dicari atau tidak. Langkah pertama BFS akan mengecek node pada level 1 yaitu node Z, S, dan T. Akan tetapi level 1 tidak memiliki node tujuan. Maka BFS akan lanjut mengecek node pada level kedua, yaitu O, O, F, R, L. Tetapi seperti pada level sebelumnya, pada level kedua ini BFS juga belum menemukan node yang dicari. Sehingga BFS akan melanjutkan pencarian ke node pada level ketiga. Level ketiga memiliki node yang dicari atau goals, setelah menemukan node yang dicari, BFS akan mengurutkan jalur graph menuju ke node yang pertama. Alhasil ditemukanlah jalur **A, S, F, B** yang merupakan jalur terdekat dari node A ke B.

Jika algoritma tersebut diterapkan kedalam bahasa pemrograman python, maka bentuk algoritma tersebut akan menjadi seperti ini :

```

graph = {
    'A': ['Z', 'S', 'T'],
    'Z': ['O'],
    'O': ['S1'],
    'S1': ['F', 'R'],
    'F': ['B'],
    'B': [],
    'R': [],
    'S': ['O1', 'F1', 'R1'],
    'O1': [],
  
```

```
'F1': ['B1'],
'B1': [],
'R1': ['C', 'P'],
'C': [],
'P': [],
'T': ['L'],
'L': ['M'],
'M': ['D'],
'D': []
}
```

```
visited = [] # list untuk node yang telah dikunjungi
queue = [] # inisialisasi antrian
```

```
def bfs(visited, graph, node):
    visited.append(node)
    queue.append(node)

    while queue:
        m = queue.pop(0)
        print(m, end="\n")

        for neighbour in graph[m]:
            if neighbour not in visited:
                visited.append(neighbour)
                queue.append(neighbour)

# kode menjalankan aplikasi
print("Following is the Breadth-First Search")
bfs(visited, graph, 'A')
```

Penjelasan :

- Pada baris pertama, graf diinisialisasi dengan menggunakan variabel, dimana setiap node merepresentasikan key dan list yang terkait merepresentasikan node yang selevel dengan simpul tersebut.
- Kemudian pada baris kedua dan ketiga, list visited dan queue diinisialisasi sebagai list kosong.
- Pada baris keempat hingga keenam, fungsi bfs() didefinisikan dengan tiga parameter yaitu visited, graph, dan node. Parameter visited merupakan list untuk menyimpan node-node yang telah dikunjungi, graph merupakan representasi dari graf menggunakan dictionary, dan parameter node merupakan simpul awal yang akan dikunjungi.
- Pada baris ke-delapan dan kesembilan, node awal dimasukkan ke dalam list visited dan queue.
- Kemudian pada baris kesepuluh, perulangan while digunakan untuk memproses queue hingga kosong.
- Pada baris kesebelas, node yang berada di awal queue dipop dan di-print.
- Pada baris ke-13 hingga ke-15, loop digunakan untuk menelusuri node-node yang bersebelahan dengan simpul m, yaitu simpul yang telah dipop dari queue. Jika simpul bersebelahan belum pernah dikunjungi, maka simpul tersebut dimasukkan ke dalam list visited dan queue.
- Pada baris ke-19 hingga ke-20, fungsi bfs() dipanggil untuk menjalankan algoritma BFS, dimulai dari simpul L.

Hasil dari kode python diatas adalah sebagai berikut :

```
Following is the Breadth-First Search
A
Z
S
T
O
O1
F1
R1
L
S1
B1
C
P
M
F
R
D
B
```

Kode python diatas dijalankan dengan menggunakan node A sebagai titik awal pencarian untuk mencari node B. Langkah langkah yang dilakukan algoritma BFS ketika mencari node B yaitu :

1. Pada langkah pertama BFS akan mencetak node A sebagai node awal pencarian, karena tidak menemui node B, maka BFS lanjut turun ke level pertama
2. Di level pertama ini BFS mencari node B dengan mengecek apakah setiap node pada level pertama = B. BFS akan mencetak ke terminal / cmd setiap node yang telah ia cek, pada level pertama ini node yang dicetak adalah Z, S, T
3. Kemudian karena tidak menemukan node B pada level pertama, BFS melanjutkan pencarian ke level kedua, yaitu pada node O, O1, F1, R1, L. Node pada level kedua ini memiliki index sebagai pembeda antara node yang memiliki value sama. Pada langkah ini BFS hanya mencetak node yang telah dilewati saja, karena belum menemukan node tujuan
4. Kemudian BFS melanjutkan pencarian menuju level ketiga, dan disini BFS juga hanya mencetak node S1, B1, C, P, M dan melanjutkan pencarian ke level 4
5. Pada level 4 BFS juga hanya mencetak node F, R, D karena tidak menemukan node B sebagai tujuan pencarian
6. Kemudian setelah turun ke level kelima, BFS menemukan node yang dicari yaitu node B. Setelah node tujuan ditemukan BFS menghentikan perulangannya dan menampilkan seluruh node yang telah dicetak kedalam terminal / cmd

DAFTAR PUSTAKA

<https://www.trivusi.web.id/2022/05/apa-itu-algoritma-breadth-first-search.html#:~:text=Breadth-first%20search%20%28BFS%29%20atau%20Breadth-first%20traversal%20adalah%20algoritma,node%20dari%20suatu%20struktur%20data%20tree%20atau%20graph.>

<https://www.edureka.co/blog/breadth-first-search-algorithm/>

<https://medium.com/@vatsalunadkat/advantages-and-disadvantages-of-ai-algorithms-d8fb137f4df2>

<https://www.youtube.com/@gunadarmacollaboration9360>