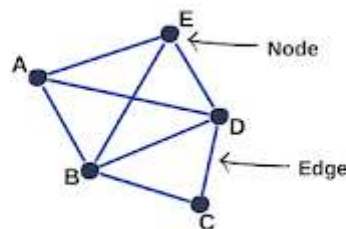


SHORTEST PATH ALGORITHM

BREADTH FIRST SEARCH

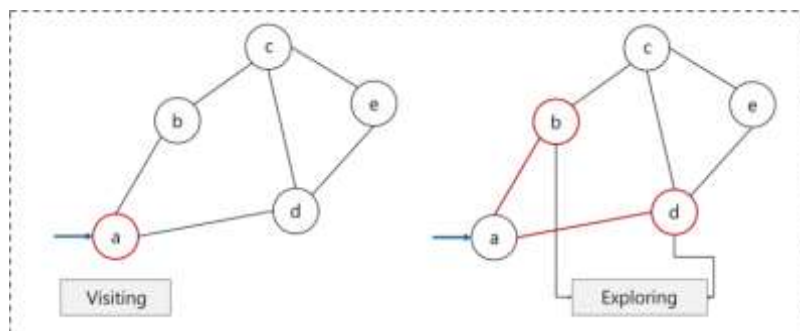
Breadth first search adalah algoritma traversing yang digunakan untuk melintasi atau mencari semua simpul atau node dari suatu struktur data tree atau graph.

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.

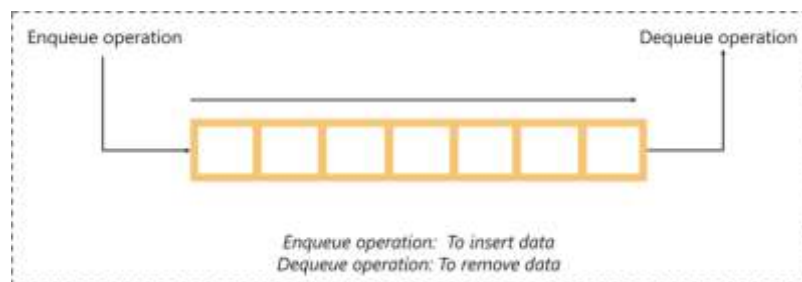


1. Visiting: mengunjungi sebuah node berarti mengunjungi atau memilih sebuah node
2. Exploring: menjelajahi node yang berdekatan (node tetangga) dari node yang dipilih

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.

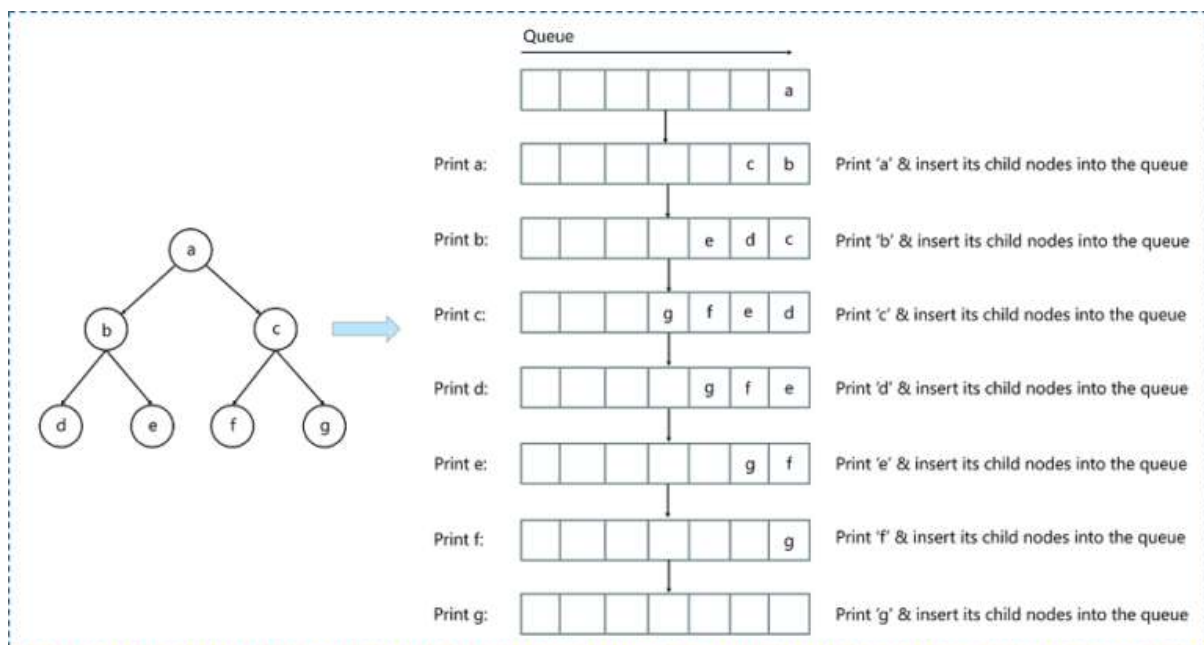
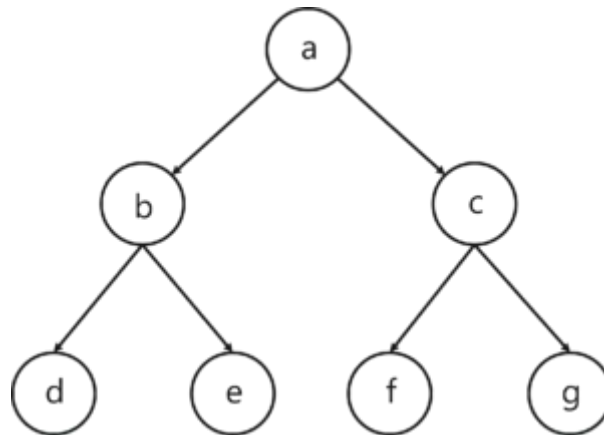
Cara kerja breadth first search:

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)



- 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

Contoh dari Breadth First Search



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.

Implementasi

```
import collections

def bfs(graph, root):

    visited, queue = set(), collections.deque([root])
    visited.add(root)

    while queue:

        # Dequeue a vertex from queue
        vertex = queue.popleft()
        print(str(vertex) + " ", end="")

        # If not visited, mark it as visited, and
        # enqueue it
        for neighbour in graph[vertex]:
            if neighbour not in visited:
                visited.add(neighbour)
                queue.append(neighbour)

if __name__ == '__main__':
    graph = {0: [1, 2], 1: [2], 2: [3], 3: [1, 2]}
    print("Following is Breadth First Traversal: ")
    bfs(graph, 0)
```

Following is Breadth First Traversal:
0 1 2 3