

TUGAS RESUME

Nama : Muhammad Irsyad

Kelas : S1IF10B

NIM : 2211102048

1. Konsep Graph

Dalam ilmu komputer dan matematika diskrit, konsep graf merupakan komponen penting. Dalam teori jaringan, optimisasi, teori graf, dan pemodelan sistem kompleks, graf sering digunakan untuk memodelkan dan menganalisis berbagai situasi dan masalah. Ini mewakili hubungan antara objek atau entitas dalam bentuk simpul (node) yang terhubung oleh sisi (edge).

Berikut adalah beberapa elemen penting dari gagasan grafik:

- a. **Simpul (Node):** Sebuah entitas atau objek khusus yang diwakili oleh titik atau lingkaran pada grafik disebut simpul. Dalam kebanyakan kasus, setiap simpul memiliki label atau identitas khusus yang membedakannya dari simpul lainnya. Dalam graf jaringan komputer, simpul dapat berupa perangkat, sedangkan dalam graf sosial, simpul dapat berupa orang.
- b. **Sisi (Edge):** Sisi graf menunjukkan adanya hubungan atau koneksi antara dua simpul. Sisi bisa memiliki arah (disebut sisi berarah) atau tidak. Selain itu, sisi dapat memiliki fitur atau bobot yang menunjukkan hubungan yang terjadi antara simpul-simpul tersebut.

2. Contoh Penerapan Graph Sehari-hari

- a. **Jaringan Sosial:** Graf digunakan untuk menganalisis hubungan sosial individu dalam jaringan sosial seperti Facebook, Twitter, dan LinkedIn. Setiap pengguna dapat digambarkan sebagai simpul, dan hubungan atau persahabatan pengguna dapat digambarkan sebagai sisi. Mengidentifikasi pengaruh, mengoptimalkan penyebaran informasi, atau menganalisis hubungan antar kelompok dapat dilakukan dengan menggunakan data tentang struktur jaringan sosial ini.
- b. **Transportasi:** Untuk perencanaan rute dan jaringan transportasi, grafik digunakan. Kota atau negara dapat digambarkan sebagai simpul, dan jalan atau rute antara lokasi tersebut digambarkan sebagai sisi. Graf dapat digunakan untuk mengoptimalkan jaringan transportasi umum, memodelkan aliran lalu lintas, atau menemukan rute terpendek antara dua lokasi.

3. Reperensentasi Graph

a. Kekuatan Matriks Adjasi:

Bagan dapat direpresentasikan menggunakan matriks dua dimensi, dengan baris dan kolom yang mewakili simpul bagan. Jika ada sisi antara simpul i dan j , maka elemen matriks baris i dan kolom j memiliki nilai 1 atau mengandung bobot/nilai yang mewakili sisi tersebut. Jika tidak ada sisi antara simpul i dan j , elemen matriksnya adalah 0 atau mengandung nilai yang menunjukkan tidak ada sisi.

b. Presentasi daftar Adjasi:

Grafik juga dapat diwakili oleh daftar yang menyimpan informasi tetangga untuk setiap simpul. Setiap node memiliki daftar node yang terhubung langsung ke node tersebut.

c. Representasi matriks lingkungan:

Bagan juga dapat direpresentasikan menggunakan matriks dua dimensi, dengan baris dan kolom yang mewakili simpul dari bagan. Elemen matriks pada baris i dan kolom j berisi nilai yang menunjukkan jumlah sisi yang menghubungkan titik i dan j .

d. Representasi dalam mode penyimpanan objek:

Grafik dapat diwakili oleh objek yang mewakili simpul dan tepi. Setiap objek simpul berisi informasi tentang simpul itu sendiri dan objek samping berisi informasi tentang tepi yang menghubungkan dua simpul.

4. Tipe Tipe Graph

a. Graf berarah dan tak berarah berbeda. Graf berarah memiliki sisi yang memiliki arah atau orientasi, yang berarti hubungan antara dua titik memiliki kejelasan arah. Sebaliknya, graf tak berarah tidak memiliki sisi yang memiliki arah, dan hubungan antara dua titik bersifat saling mengarah.

b. Graf Berbobot dan Tanpa Bobot: Graf berbobot adalah graf di mana setiap sisi memiliki bobot atau nilai numerik yang berkaitan dengan hubungannya. Ini biasanya digunakan untuk menunjukkan jarak, biaya, kekuatan, atau atribut lainnya yang relevan antara dua titik. Graf tanpa bobot adalah graf di mana sisi-sisinya tidak memiliki bobot.

c. Graf terhubung dan tidak terhubung berbeda. Graf terhubung memiliki setidaknya dua komponen yang tidak dapat dicapai satu sama lain. Dalam graf terhubung, setiap simpul dapat dicapai dari simpul lainnya melalui rangkaian sisi.

- d. Graf siklik dan asiklik berbeda. Graf siklik memiliki siklus, yaitu serangkaian sisi yang membentuk jalur kembali ke simpul awal.
- e. Graf Lengkap: Sebuah graf lengkap memiliki sisi yang menghubungkan setiap pasangan simpul. Dalam graf lengkap dengan n simpul, ada $n(n-1)/2$ sisi.

5. Operasi dan Algoritma Graph

Fungsi dan algoritma berikut biasanya digunakan dalam bagan:

a. Penelusuran (Jelajahi):

Pencarian kedalaman pertama (DFS):

- Selesaikan eksplorasi bagan lengkap dengan mengunjungi sebanyak mungkin titik sudut sebelum kembali.

Pencarian luas-pertama (BFS):

- Lakukan eksplorasi grafis yang ekstensif, pertama-tama kunjungi semua titik yang berdekatan sebelum berpindah ke titik yang lebih jauh.

b. Pohon rentang minimum:

Algoritma Kruskal:

- Menemukan pohon rentang minimum dalam graf berbobot tidak berarah dengan memilih sisi terkecil berdasarkan bobotnya.

Algoritma Prim:

- Temukan pohon merentang minimum dalam graf berbobot tidak berarah dengan memilih simpul terkecil berdasarkan bobot sisi yang menghubungkannya.

c. Jalur terpendek:

Algoritma Dijkstra:

- Temukan jalur terpendek antara dua simpul dalam diagram bobot tak berarah atau berarah dengan bobot positif.

Algoritma Bellman-Ford:

- Temukan jalur terpendek antara dua simpul dalam graf berbobot tak berarah atau berarah dengan bobot yang bisa berupa angka positif atau negatif.

d. Penyortiran topologi:

Penyortiran topologi:

- Mengurutkan simpul dari graf berarah sehingga setiap sisi mengarah dari simpul dengan indeks lebih rendah ke simpul dengan indeks lebih tinggi.

Komponen yang terhubung:

- Komponen yang terhubung:
Temukan dan kelompokkan simpul-simpul dari suatu graf menjadi komponen-komponen yang terhubung, di mana setiap simpul dari suatu komponen dapat menjangkau simpul-simpul lain dari komponen tersebut.

e. Deteksi episode:

Deteksi episode:

- Memeriksa apakah grafik memiliki periode, apakah itu grafik berarah atau tidak berarah. Aliran Maksimum:

Algoritme Ford–Fulkerson:

- Temukan aliran maksimum antara dua simpul dalam grafik berarah menggunakan jalur terpendek yang masih dapat menerima aliran tambahan.

6. Implementasi Grap Pada C++

```

#include <iostream>

#include<list>

using namespace std;

class graph{
public:
    list<int> *adjlist;

    int n;

    graph(int v){
        adjlist=new list<int> [v];

        n=v;
    }

    void addedge(int u,int v,bool bi){
        adjlist[u].push_back(v);

        if(bi){
            adjlist[v].push_back(u);
        }
    }

    void print(){
        for(int i=0;i<n;i++){
            cout<<i<<"-->";

            for(auto it:adjlist[i]){
                cout<<it<<" ";
            }

            cout<<endl;
        }

        cout<<endl;
    }
};

int main() {
    graph g(5);

    g.addedge(1,2,true);
    g.addedge(4,2,true);
    g.addedge(1,3,true);
    g.addedge(4,3,true);
    g.addedge(1,4,true);

    g.print();
}

```

Pada contoh di atas, struktur data bagan diimplementasikan menggunakan representasi daftar yang berdekatan. Kelas Grafik memiliki metode `addEdge()` untuk menambahkan tepi ke grafik. Selain itu, ada metode `BFS()` yang mengimplementasikan algoritma pencarian luas-pertama untuk melakukan penjelajahan bagan luas.

Dalam fungsi `main()`, grafik dengan 4 titik (0, 1, 2, 3) dibuat. Tepi ditambahkan menggunakan metode `addEdge()`. Kemudian metode `BFS()` dipanggil dan polling dimulai pada node 2. Output dari program di atas adalah mencetak traversal BFS mulai dari node 2: 2 0 3 1