Bing

Saya dapat menjelaskan kembali algoritma Dijkstra yang dioptimasi heap dalam buku Pemrograman Kompetitif Dasar. Berikut adalah penjelasannya:

- **Algoritma Dijkstra**: Algoritma ini adalah algoritma yang digunakan untuk mencari jalur terpendek dari satu titik ke semua titik lain dalam sebuah graf berbobot. Algoritma ini bekerja dengan cara memilih simpul dengan jarak terdekat dari simpul awal yang belum dikunjungi, dan mengupdate jarak ke simpul-simpul yang dapat dijangkau melalui simpul tersebut. Algoritma ini dapat dioptimasi dengan menggunakan struktur data priority queue, yang dapat mempercepat pencarian simpul dengan jarak terdekat.
- **Heap**: Heap adalah salah satu implementasi dari priority queue, yaitu struktur data yang menyimpan elemen-elemen berdasarkan prioritasnya, sehingga elemen dengan prioritas tertinggi dapat diambil atau dihapus dalam waktu singkat. Heap adalah pohon biner yang seimbang dan memenuhi sifat heap, yaitu nilai setiap simpul tidak lebih besar dari nilai anak-anaknya. Dengan menggunakan heap, operasi menambahkan dan menghapus elemen dari priority queue dapat dilakukan dalam waktu O(log n), di mana n adalah jumlah elemen dalam priority queue.
- **Algoritma Dijkstra yang dioptimasi heap**: Algoritma ini adalah modifikasi dari algoritma Dijkstra yang menggunakan heap sebagai struktur data untuk menyimpan simpul-simpul yang belum dikunjungi. Algoritma ini memiliki langkah-langkah sebagai berikut:
    - Buat sebuah heap Q yang berisi semua simpul dalam graf, dengan prioritas berdasarkan jarak ke simpul awal. Inisialisasi jarak semua simpul menjadi tak hingga, kecuali simpul awal yang diinisialisasi menjadi nol.
    - Selama Q tidak kosong, lakukan hal berikut:
        - Ambil dan hapus simpul u dari Q yang memiliki prioritas tertinggi (jarak terkecil).
        - Untuk setiap tetangga v dari u yang masih ada dalam Q, lakukan hal berikut:
            - Hitung jarak alternatif alt sebagai jumlah jarak u dan bobot sisi (u, v).
            - Jika alt lebih kecil dari jarak v saat ini, maka update jarak v menjadi alt dan simpan u sebagai simpul sebelumnya dari v.
            - Masukkan kembali v ke Q dengan prioritas baru berdasarkan jarak v yang baru.
    - Kembalikan jarak dan simpul sebelumnya dari semua simpul sebagai hasil.

Untuk lebih jelasnya, Anda dapat melihat contoh kode program algoritma Dijkstra yang dioptimasi heap dalam bahasa Python pada link ini atau dalam bahasa C++ pada link ini. Anda juga dapat membaca penjelasan lebih detail tentang algoritma Dijkstra dan heap pada link ini atau link ini. Semoga bermanfaat! 😊