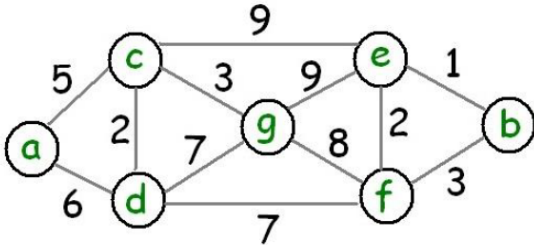


Latihan Soal Algoritma Greedy

(diambil dari soal-soal UTS)

Soal 1:

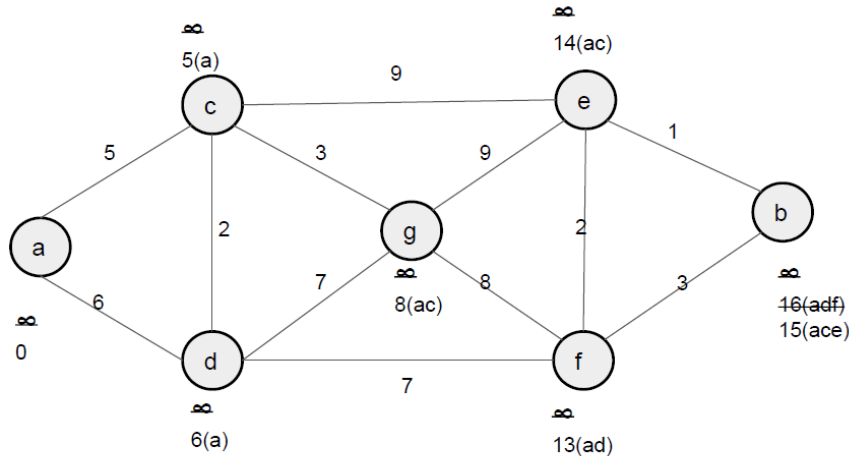
Diberikan graf berikut ini, kita akan menentukan lintasan terpendek dari simpul a ke semua simpul lainnya.



- Gunakanlah algoritma Kruskal untuk menentukan lintasan terpendek tersebut. Sebelum mengerjakan, tuliskanlah strategi greedy yang digunakan Kruskal. **(Nilai 7.5)**
- Gunakanlah algoritma Dijkstra untuk menentukan lintasan terpendek tersebut. Sebelum mengerjakan, tuliskanlah strategi greedy yang digunakan Dijkstra. **(Nilai 10)**
- Berikanlah kesimpulan dari hasil (a) dan (b). **(Nilai 2.5)**

Jawaban:

Solusi Soal a: Dijkstra (Nilai 10)



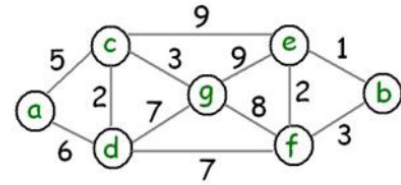
Lintasan terpendek Dijkstra:

a-b: a-c-e-b = 15; a-c: a-c = 5;

a-d: a-d = 6

a-e: a-c-e = 14; a-f: a-d-f = 13;

a-g: a-c-g = 8

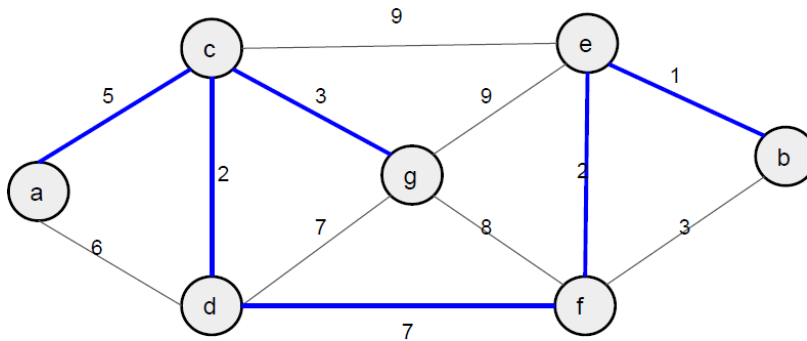


Strategi greedy

Pada setiap langkah, pilih simpul yang belum terpilih dan memiliki panjang lintasan terpendek dari simpul awal, lalu update simpul lain yang belum terpilih dengan $L(u) + G(u,v)$ jika $L(u) > L(u) + G(u,v)$.

(versi slide kuliah): Pada setiap langkah, ambil sisi yang berbobot minimum yang menghubungkan sebuah simpul yang sudah terpilih dengan sebuah simpul lain yang belum terpilih.

Solusi b: Kruskal (nilai 7.5)



Strategi greedy:

Pada setiap langkah, pilih sisi dari graf G yang mempunyai bobot minimum tetapi tidak membentuk sirkuit di T .

Lintasan terpendek:

a-b: a-c-d-f-e-b = 17

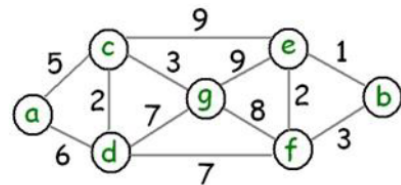
a-c: a-c = 5

a-d: a-c-d = 7

A-e: a-c-d-f-e = 16

A-f: a-c-d-f = 14

A-g: a-c-g = 8



Solusi c (nilai 2.5)

Lintasan terpendek Dijkstra:

a-b: a-c-e-b = 15;

6

a-e: a-c-e = 14;

Lintasan terpendek Kruskal:

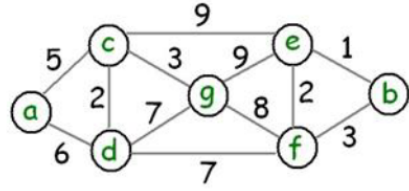
a-b: a-c-d-f-e-b = 17;

= 7

a-e: a-c-d-f-e = 16;

Kesimpulan:

Dijkstra menjamin memberikan lintasan terpendek ke semua simpul; sedangkan Kruskal hanya memberikan pohon merentang minimum yang tidak memberikan lintasan terpendek ke semua simpul.



a-c: a-c = 5;

a-f: a-d-f = 13;

a-c: a-c = 5;

a-f: a-c-d-f = 14;

a-d: a-d =

a-g: a-c-g = 8

a-d: a-c-d

a-g: a-c-g = 8

Soal 2:



Pada persoalan *muddy city*, terdapat suatu kota tanpa jalan aspal. Jika hujan badai, jalanan menjadi sangat berlumpur, mobil terjebak di lumpur dan boots orang menjadi kotor. Walikota memutuskan untuk memasang *paving block* lebar, tetapi hanya ingin menghabiskan biaya seminimal mungkin. Terdapat dua kondisi: (1) *Paving block* cukup dipasang sehingga penduduk dapat datang

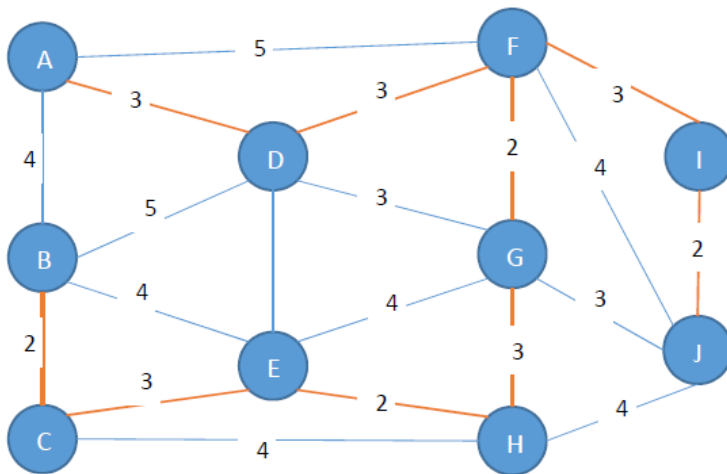
dari rumah mereka ke rumah lain melalui jalan ber-*paving block*, mungkin saja harus melalui rumah lainnya; (2) Jumlah *paving block* seminimal mungkin agar biayanya juga minimal. Bantulah Pak Walikota dengan memberikan solusi jalur *paving block* yang akan dipasang. (Sumber gambar:

<http://statklee.github.io/website-csunplugged>)

1. a. (Nilai 5) Representasikanlah gambar kota di atas menjadi sebuah graf, dengan simpul merepresentasikan rumah, dan bobot sisi merepresentasikan jumlah *paving block* yang dibutuhkan. Jembatan dihitung sebagai satu *paving block*.
b. (Nilai 15) Jika diselesaikan dengan *exhaustive search*, jelaskan strateginya seperti apa (tidak perlu pseudo-code), lalu tentukanlah kompleksitas algoritmanya dalam notasi big O.
2. (Nilai 15) Jika diselesaikan dengan *greedy*, rancanglah strategi *greedy* terbaiknya, dan tentukanlah kompleksitas algoritmanya dalam notasi big O.

Jawaban:

- a. (Nilai 5) Representasikanlah gambar kota di atas menjadi sebuah graf, dengan simpul merepresentasikan rumah, dan bobot sisi merepresentasikan jumlah *paving block* yang dibutuhkan. Jembatan dihitung sebagai satu *paving block*.



b. (Nilai 15) Jika diselesaikan dengan exhaustive search, jelaskan strateginya seperti apa (tidak perlu pseudo-code), lalu tentukanlah kompleksitas algoritmanya dalam notasi big O.

Persoalan ini adalah persoalan minimum spanning tree.

Alternatif 1 strategi:

- enumerasi semua spanning tree berupa lintasan hamilton dgn cara membuat semua permutasi semua simpul (telah memenuhi syarat tidak siklik), terdapat $(n-1)!$
- evaluasi setiap spanning tree dengan menghitung bobot lintasan
- Ambil spanning tree dengan bobot paling minimum

Kompleksitas: $O(n.n!)$, n adalah jumlah simpul

Alternatif 2 strategi:

- enumerasi semua subset sisi, terdapat 2^n , dengan n adalah jumlah sisi.
- evaluasi setiap subset yang memenuhi spanning tree (tidak siklik, melewati semua simpul) dengan menghitung bobot lintasan
- Ambil spanning tree dengan bobot paling minimum

Kompleksitas: $O(n.2^n)$, n adalah jumlah sisi

2. (Nilai 15) Jika diselesaikan dengan greedy, rancanglah strategi greedy terbaiknya, dan tentukanlah kompleksitas algoritmanya dalam notasi big O.

Alternatif 1: dgn menggunakan Kruskal:

1. Sort semua sisi berdasarkan jumlah paving block: BC(2), EH(2), FG(2), IJ(2), AD(3),...
2. Tambahkan sisi secara berurutan untuk membentuk pohon jika tidak membentuk siklik. Jika membentuk siklik, sisi diabaikan. Hal ini dilakukan sampai semua simpul sudah dicapai.

Solusi optimum : BC(2)-EH(2)-FG(2)-IJ(2)-AD(3)-CE(3)-DF(3)-FI(3)-GH(3). Total bobot=23.

Kompleksitas: $O(|E|\log|E|)$

Alternatif 2: dgn menggunakan Prim:

Ambil simpul awal (misalnya A atau simpul dgn bobot terpendek), lalu bangun tree T secara greedy, dengan menambahkan sisi terpendek yang menghubungkan salah satu simpul yang telah terpilih dengan simpul yang belum terpilih.

Solusi optimum: A-D-F-G-H-E-C-B-I-J atau B-C-E-H-G-F-D-A-I-J. Total bobot=23.

Kompleksitas: $O(n^2)$