Graf terdiri dari V (Himpunan dari verteks yang bukan himpunan kosong), E (Adalah himpunan edge dengan catatan boleh himpunan kosong), edge punya satu atau dua verteks, verteks merupakan endpoints, edges menyambungkan endpoints.

V = {a, b, c }

E = {{a},{a,b},{b,c}}

A dikatakan identik jika setiap elemen dari A adalah elemen dari B

jika menggunakan kurung buka tutup disebut oder paired

(a,b) tidak = (b,a), notasi graf

Jenis Graph

simple graph = graf yang mengoneksikan dua verteks berbeda, dan tidak ada 2 edge yang mengoneksikan dua verteks

tidak ada loop tidak ada pararel edge

loop = edge yang menghubungkan satu verteks

multiple graph = edge mengkoneksikan 2 verteks sama

menggunakan indeks tiap edge contoh e1,e2,e3,e4 karena {a,b} = {a,b}

e1 = {a,c}, e2 = {a,b}, e3 = {a,b},e4 {b,c}

graph yang tidak punya arah = undirected graph

graph yang punya arah = directed graph/digraph

bukan order pair dari verteks

order pair pake (u , v) berawal dari u berakhir di v

gambar tidak pengaruh di diskrit yang penting data

adjacent/tetangga = dua verteks dikatakan adjacent jika u dan v adalah endpoint dari edge (dihubungkan oleh edge)

degre = jumlah edge yang dihubungkan dengan dirinya (verteks), jika loop dihitung 2, sisanya 1

rumus edge = jumlah degre/2 karena edge yang sama tidak di hitung.

incident = e1 = {a,b},e1 bertepi pada a dan b

directed graph

saat (u,v) adalah edge dari graph G maka u bertetangga ke v begitu sebaliknya, jika directed graph maka hanya u ke v (sesuaikan dengan arah edge)

dan degrenya dihitung terpisah yang in dan out

Special simple graph =

complete graph = semua verteks lainnya terhubung langsung dengan vertek lainnya

cycle = graph mulai dari verteks awal berakhir di verteks awal kembali

n-cubes = verteks = 2^n (berapa bit), garis yang terhubung harus beda 1 bit

1 cubes = 1 garis

2 cubes = 4 garis (2 dimensi) (bipartite)

3 cubes = 8 garis (3 dimensi) (bipartite)

bipartite = vertex yang bisa disambungkan menjadi himpunan, setiap edge di graph bisa digabungkan

subgraph G=(v,E) adalah H= (w,f), dimana w subset v dan f subset e, subgraph g jika H tidak = G

subset tidak= proper subset

adjency matriks menggunakan titik terhubung kemana aja, dan jika dia graph maka akan simetris

graf planar : bisa digambarkan pada satu bidang tanpa adanya edge yang bersilangan

representasi planar graph adalah membagi bidang menjadi region dengan dibatasi oleh edge, dan tiap bidang tidak boleh sama warnanya antara edge yang sama

warna : resource = sumber daya = limited = minimal = reuse

pewarnaan graph : bisa mewarnai region, vertex, atau edge

2 litas grpah : jika suatu region menjadi vertex maka hasilnya akan sama seperti region tidak menjadi vertex

mewarnai vertex tidak boleh mewarnai vertex yang bertetanggaan sewarna

chromatic number : jumlah warna tekecil dari jumlah warna yang dibutuhkan

proses mewarnai : mengalokasikan resource

grafik isomorfik = Grafik sama cuman bentuk beda, tapi bentuknya bisa sama dan memili korespondensi 1-1

untuk memastikan 2 buah grafik isomorfik dengan cara memeriksa matrik ketetanggaany, jika matrik ketetanggannya sama maka kedua graf isomorfik

2 buah graf dikatakan isomorfik jika :

- jumalh simpul dan sisi pada kedua graf sama.

- jumlah simpul yang mempunyai derajat tertentu harus sama

- jika pada G1, u1 bertetangga v1 dan w1, sedangkan G2, u2 bertetangga v2 dan w2, maka derajad v1 dan v2, w1 dan w2 harus sama

- terdapat korespondensi 1-1 V1 ke V2

Connectivity

paths/lintasan : deretan edge yang berawalan dari suatu vertex atau graph yang travel dari vertex ke vertex lainnya sepanjang edge graph

contoh garis a b c = dengan edge e1 dan e2 maka lintasannya e1e2 atau a b c (tinggal milih edgenya), jika ada multi graph pilih salah satu e

undirected graph disebut terhubung jika ada lintasan dianata kedua vertex graph tesebut

pada directed graph:

strongly conected : antar vertex itu pasti ada lintasannya

weekly connected : antar vertex tidak ada arah

tree : conected undirected graph tanpa sirkuit dan loop

dari 2 vertex berbeda hanya ada 1 edge

definisi 2 : root tree : suatu vertex yang memiliki garis ke bawah (menjadi parent dan child) parent tau child tapi child belum tentu tau parentnya

pre order : 2+3\*5 = +2\*35 (operator angka operator)

post order : 2+3\*5 = 2,3,5,\*,+ (angka operato angka)

kruskal : fokus di edge

prim :fokus di vertex

Logical :

Ingkaran (~)

Konjungsi (^)

Disjungsi (v) = (~q^…)

Implikasi (->) = (~q^…)

Biimplikasi (<->)