

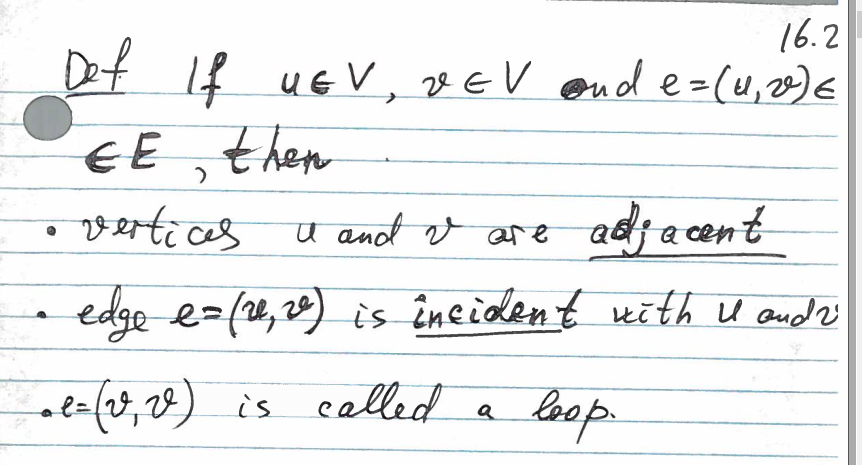
V是非空集合，E是两个V组成的Edge，因为是属于而不是等于，所以并不是每对vertice都组成edge，那么pair（v，e）就是directed graph on v//ve都是集合，或者叫做digraph

如果没有direction关系，换句话说E是一个顺序不重要的unordered pairs，那么G还是这样写，只是叫做undirected graph

不管是directed或undirected

V：vertex set of G

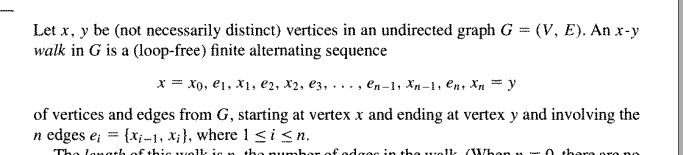
E: edge set of G



Adjacent: 如果u属于V，V属于V，而uv之间存在edge，那么这两个人就是adjacent的//临近

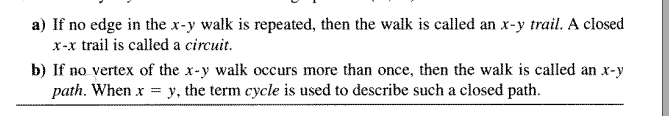
Incident: 一个edge和他任意一个端点形成Incident关系

一个自己到自己的edge叫做一个loop，



如果XY之间存在一条路径并且没有一个edge是重复的

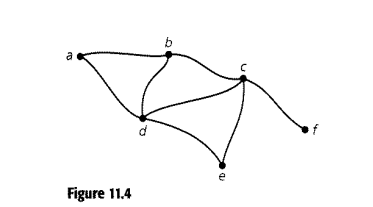
那么这个ei路径(walk)叫做x-y trail



一个closed x-x trail 叫做circuit

如果每个点出现次数只有一次或零次，那么这个walk叫做x-y,path，如果x=y,那么叫做cycle

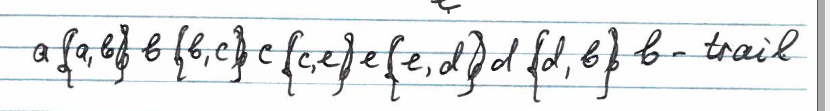
例子



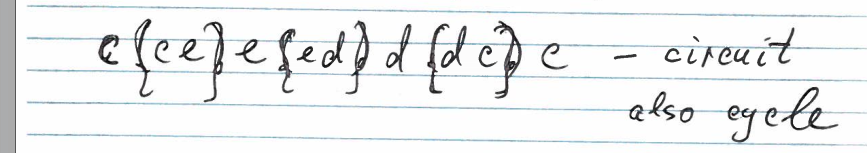
1. 

就做ab walk，因为bd db两个edge重复了（以上所有概念都是建立在undirected上的）

2.



这个叫做ab trail，但是b这个点重复了两次，不能叫做path

3. 

这里每个店出现一次，所以能叫circuit或cycle//没有edge重复叫circuit，没有vertex重复叫做cycle

总结

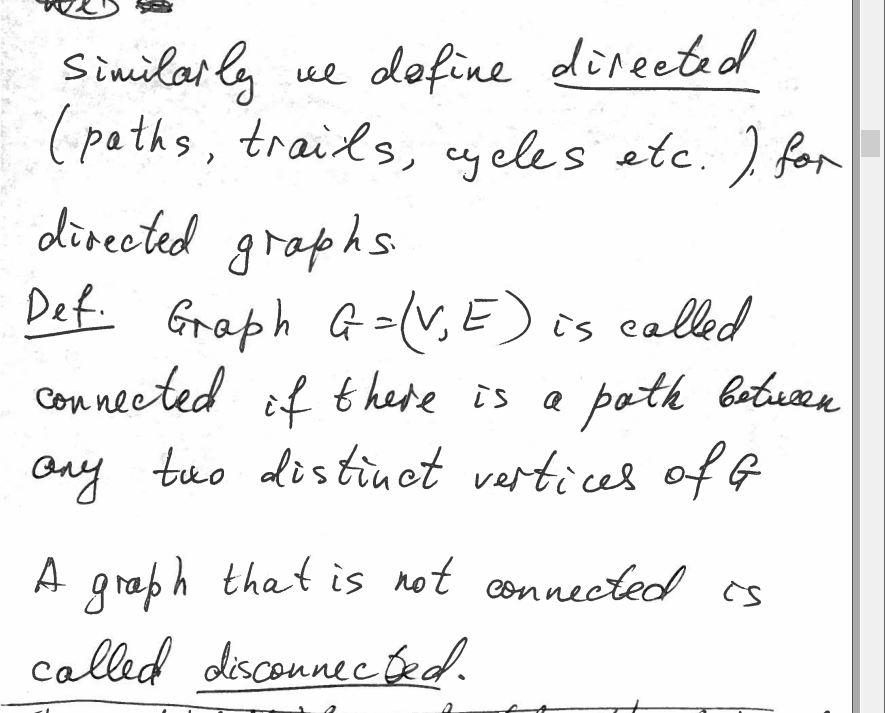
Undirected

没有edge 重复：trail, 首尾相连，circuit

没有vertex重复， path，首尾相连，cycle

一个edge （v,v）自high叫做Loop

Directed中的path, trail, cycle什么也是同一个定义

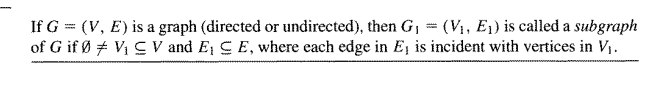


Connected:如果任意两个不同点之间都有path//没重复点

不是connected就是disconnected



如果一个undirected graph， AB点之间存在一个trail，那么ab之间必存在一个path

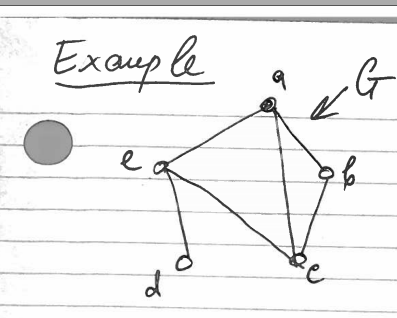
11.2

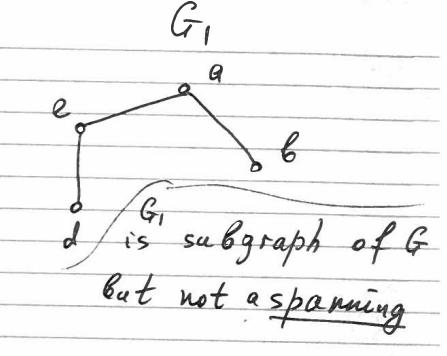


一个G为GRAPH,那么取出其中部分点与这些点组成的edge形成的新graph叫做subgraph//原先不存在的edge不能凭空生成

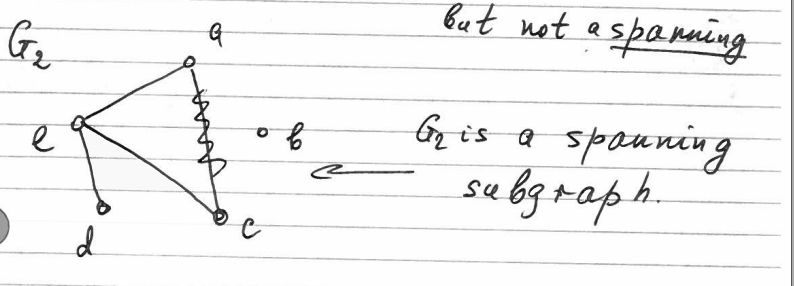
如果这个V1=V，那么G1就叫做G的spanning subgraph生成子图//edge不一定一样

原图

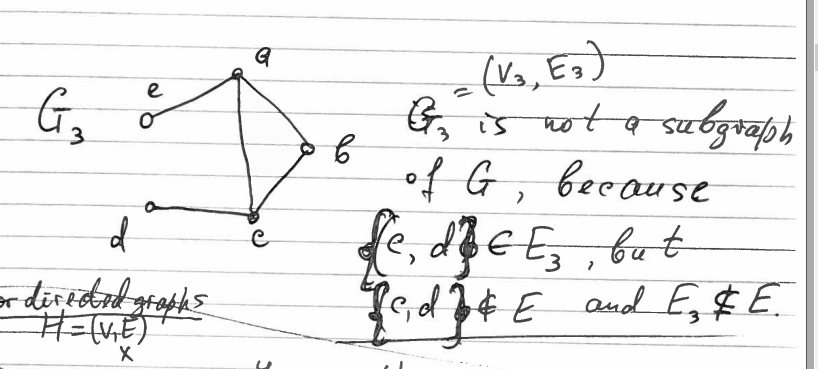




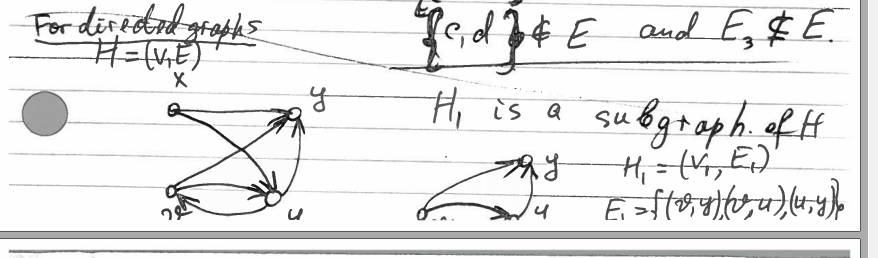
是subgraph但不是spanning ，因为点不一样，只有部分点



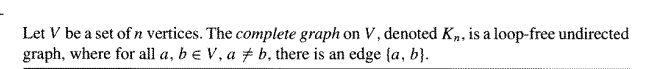
Spanning subgraph



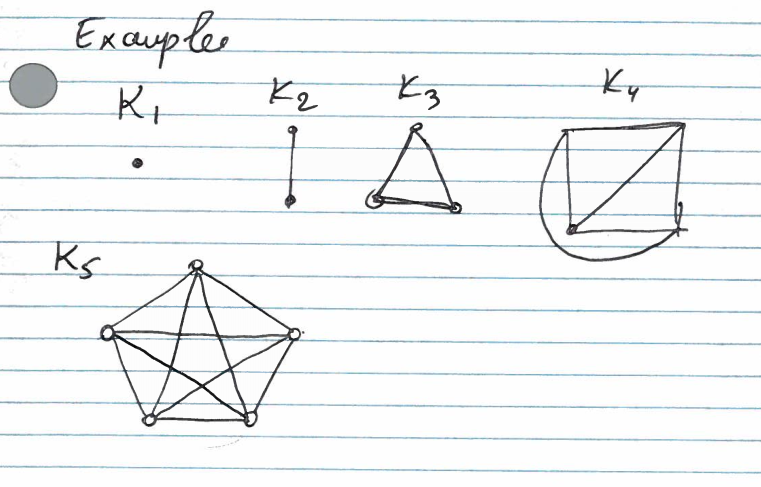
不是subgraph,因为原先没有DC edge

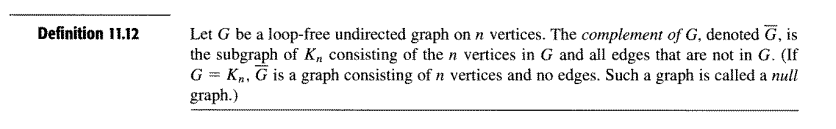


Directed graph要考虑方向

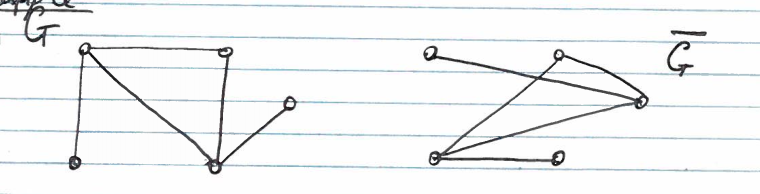


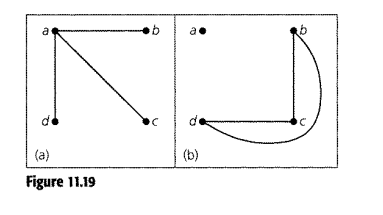
Complete graph就是一个undirected graph任意两个点之间都有edge，记做Kn

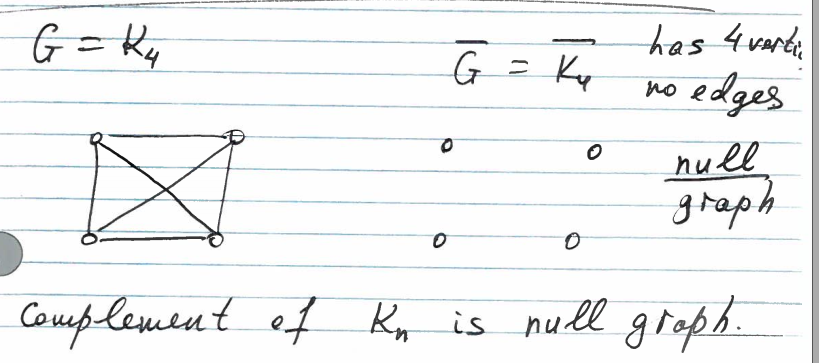




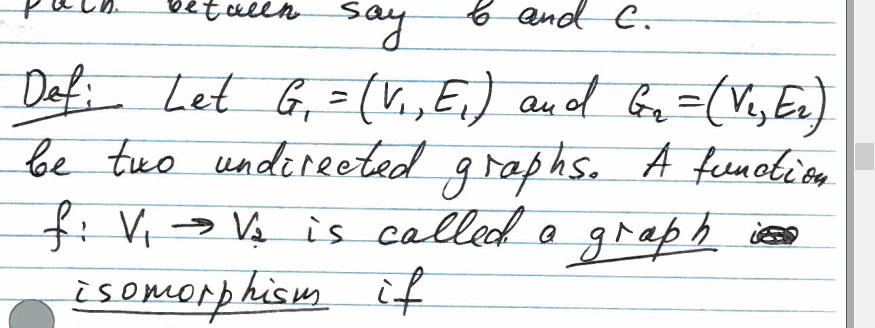
Complement 写法就是原G上面加一个杠，就是点不变，把原来不存在关系的两个edge补上

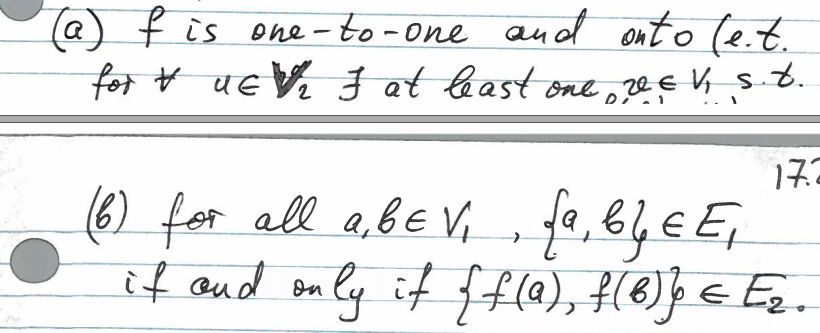






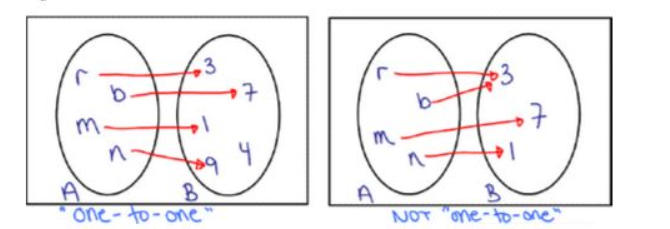
如果原G=K4，已经是compete graph了，那么complementary graph只有四个点，叫做null graph



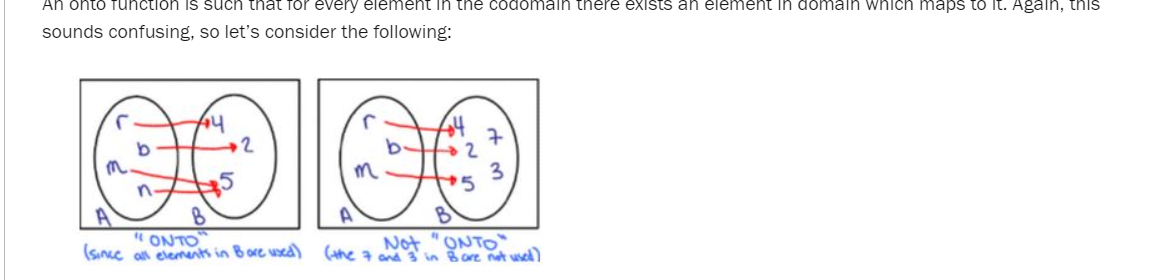


如果G1 G2两个是Undirected graph,通过一个function让V1得到V2//vertices，那么这个function叫做graph isomorphism,如果

A/f是一个one to one 和onto 关系



One to one就是一个y只有一个对应的x

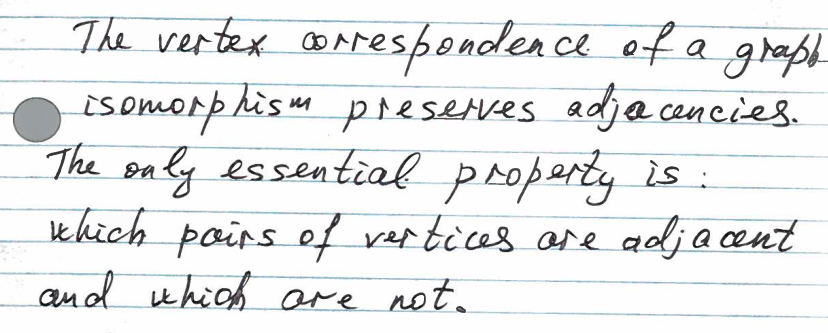


Onto就是每一个y都有一个对应的X

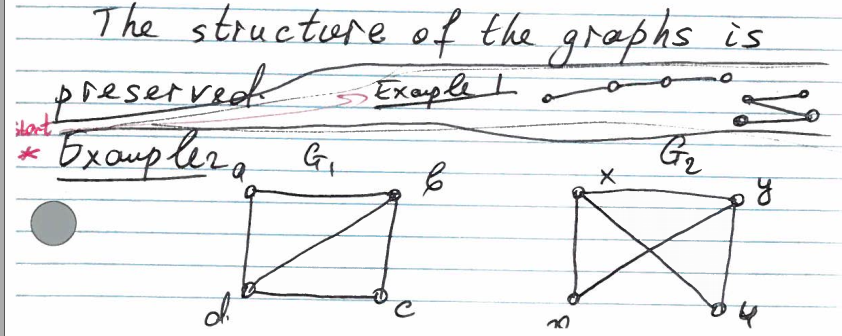
换句话说V2里的每个点都是有对应的不同X，每一个X都对应不同的点

B/如果原graph两个点之间有edge，那么新graph的两个点之间也必有edge

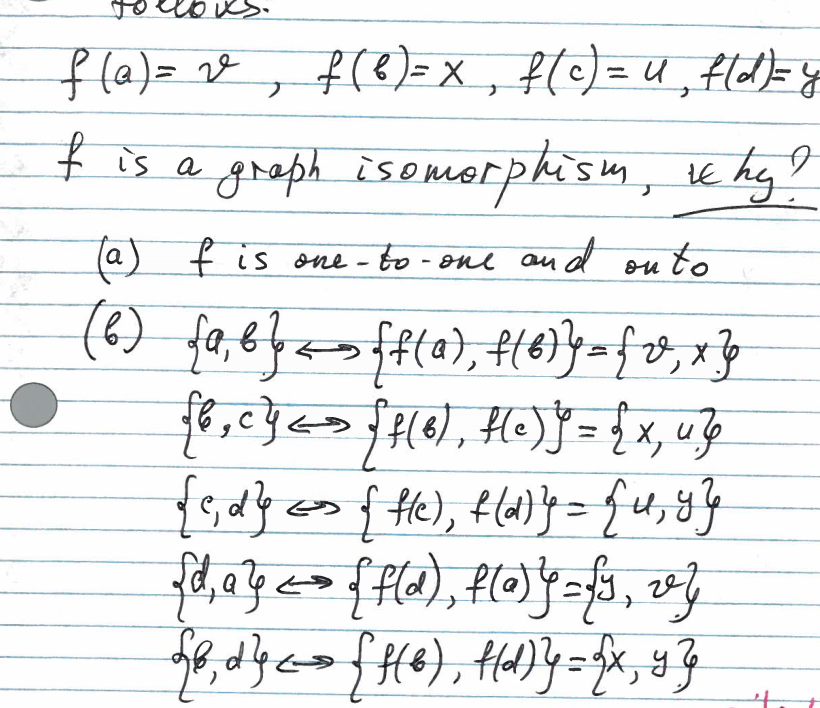
如果这样一个function存在，那么G1 G2叫做isomorphic graphs



换句话说，对应的vertex还是保持着adjacencies临近关系。所以isomorphism唯一必要条件就是知道哪些vertice是邻近的，然后挑选部分点或所有点形成新的V2



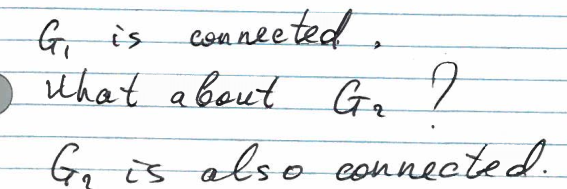
原来的结构仍然保存，f(a)=左下角，f(b)等于X，f(c)等于右下角，f(d)等于y，bd之间有关系，xy之间有关系，x右下角有关系，fbfc有关系



证明过程，

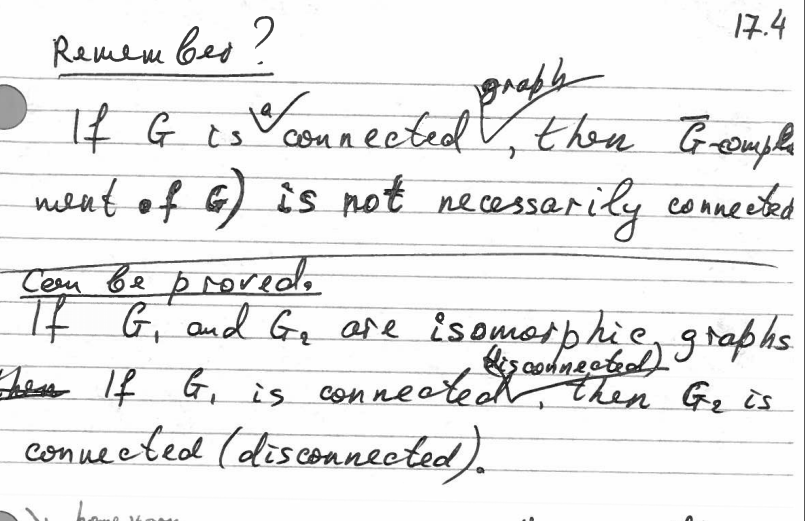
1、每个y对应一个x，满足one to one，每个 y都有对应x，满足Onto

2.然后原先的edge都存在



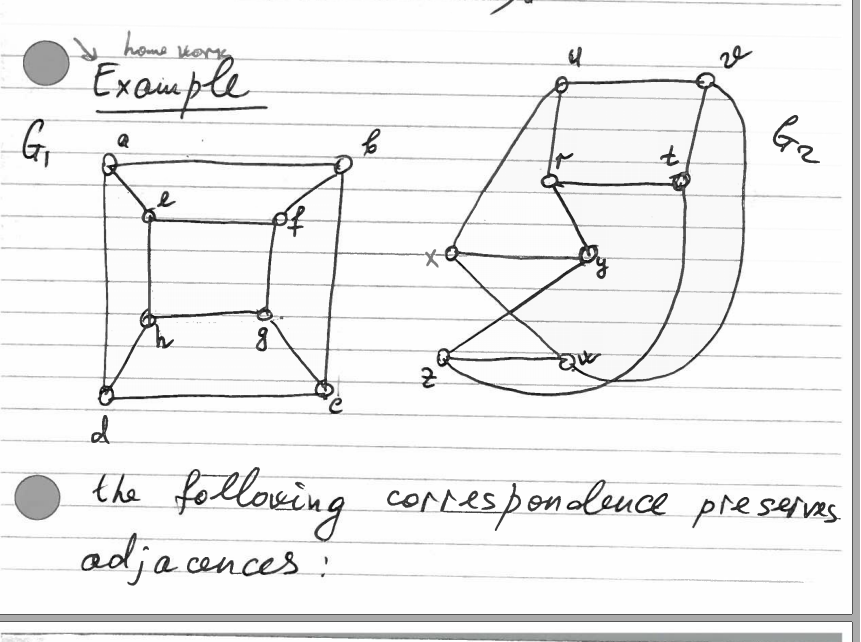
如果原先G1是connected，那么G2也将是connected

总结



如果G是connected，那么他的complementary graph不一定是connected，但如果有一个G2和他是isomorphic 关系，那么如果G1是connected，G2必是connected，反过来也成立

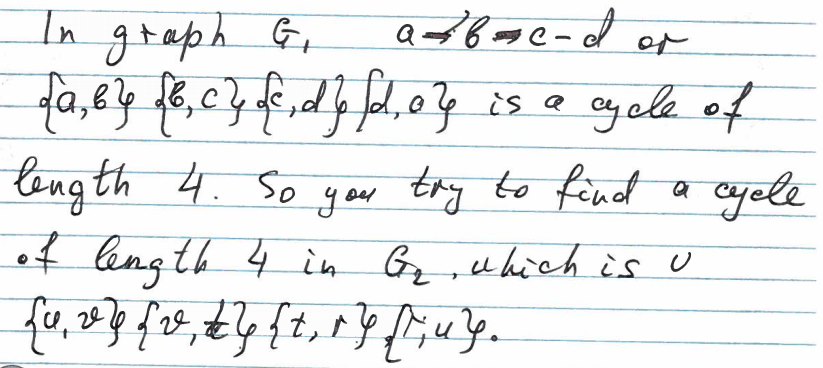
证明过程



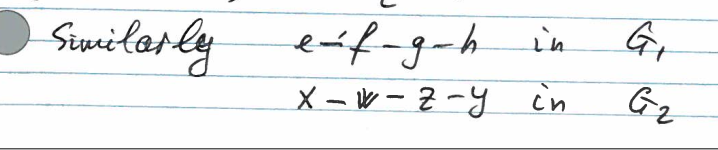
接下来的对应过程保证了adjacencies

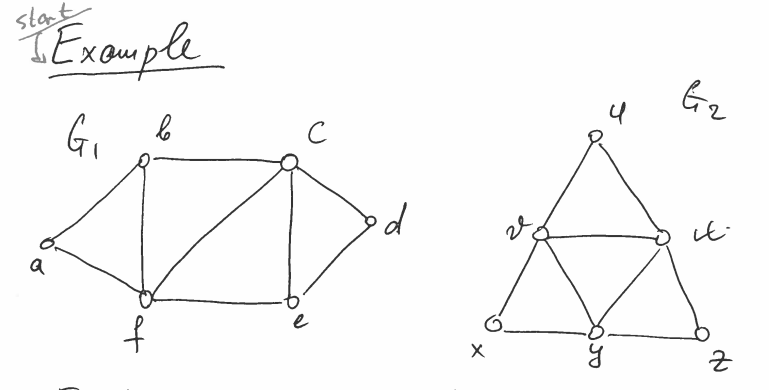


这是点的对应过程，



那么原图的任意cycle等路径都能找到对应的新cycle



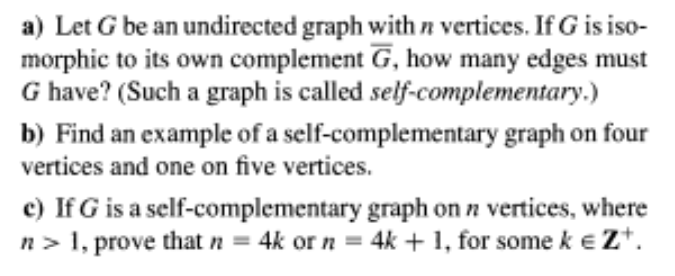


这两个isomorphic吗

不,因为a对另外两个点adjacent，那么他应该对应x u z中一个，同理d，但是没有第三个这样的点

证明isomorphic的一个主要方法就是一个点所连接的另外几个点（几根edge）

11.2 例题



a.

let edge of G=E1, edge of =E2

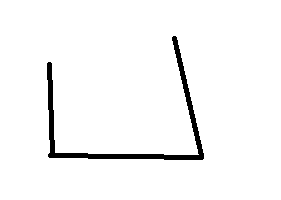
Firstly, if G and are isomorphic, then they have same numbers of edges, E1=E2

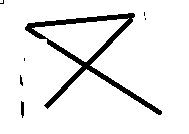
Secondly, edge of G + edge of = edge of compete graph ,E1+E2= nC2

So, E1=nC2/2= n!/(2\*(n-2!)\*2!)=n(n-1)/4

b.

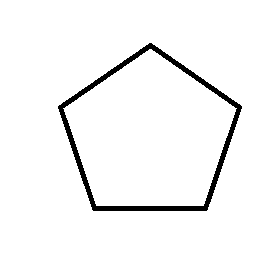
b.1: based on part a, 4\*3/4=3 , so it have 3 edges

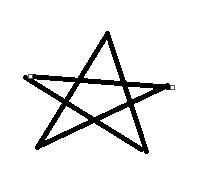
G: 2 vertices have 1 edge, 2 vertices have 2 edge

: 2 vertices have 1 edge, 2 vertices have 2 edge

True

b.2: 5\*4/4=5, it have 5 edges

G: 5 vertices have 2 edge

: 5 vertices have 2 edges

True

c.

based on part a

n(n-1)/4, n and n-1 must be one odd one even, so the even one must be the multiple of 4, then n(n-1)/4 can be integer.

So n=4k or n=4k+1 //n-1=4k