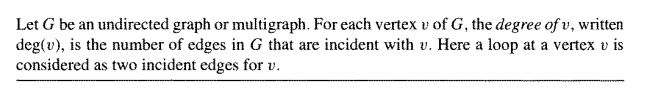
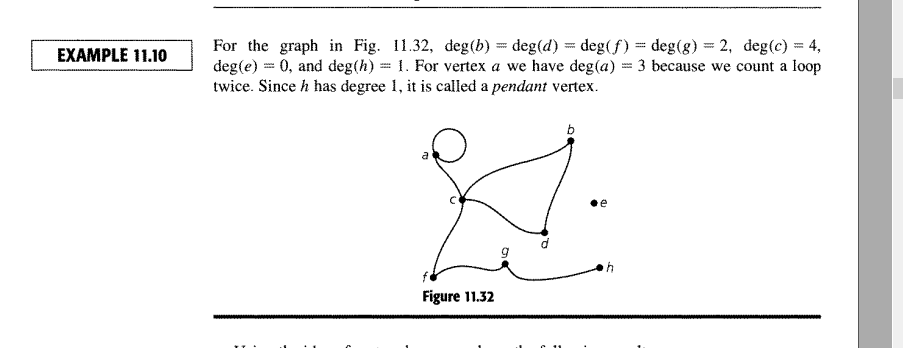
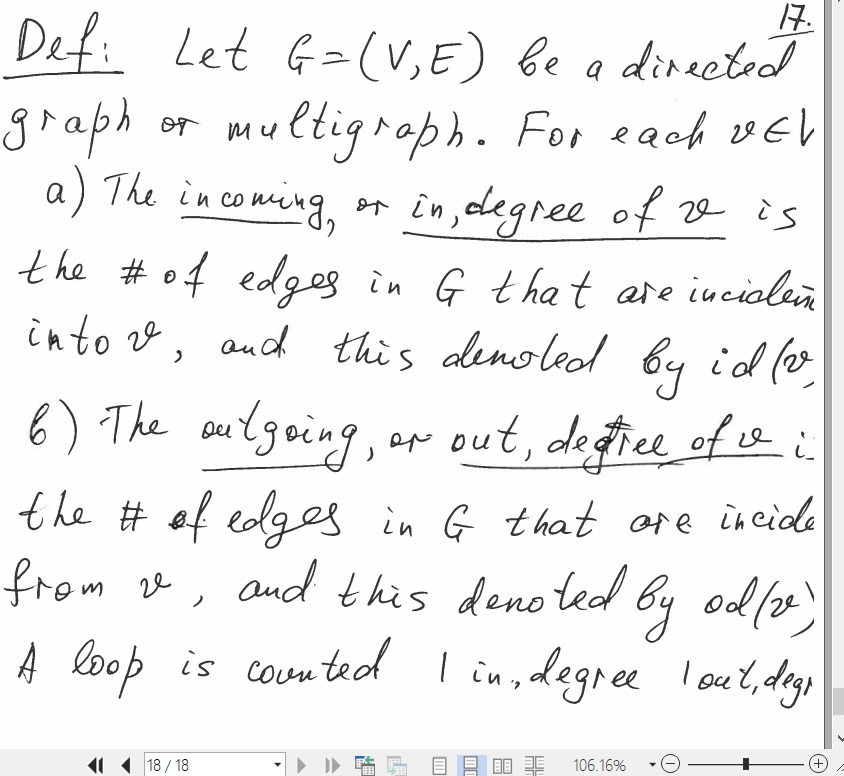
11.3



一个vertex的degree就是他连着几根edge，



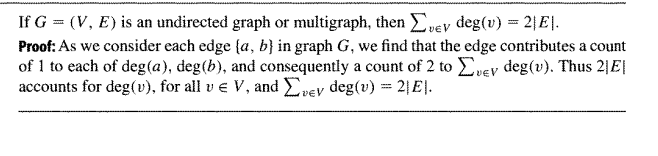
一个自我Loop算两次deg， pendant 的那个不会考



如果是directed graph

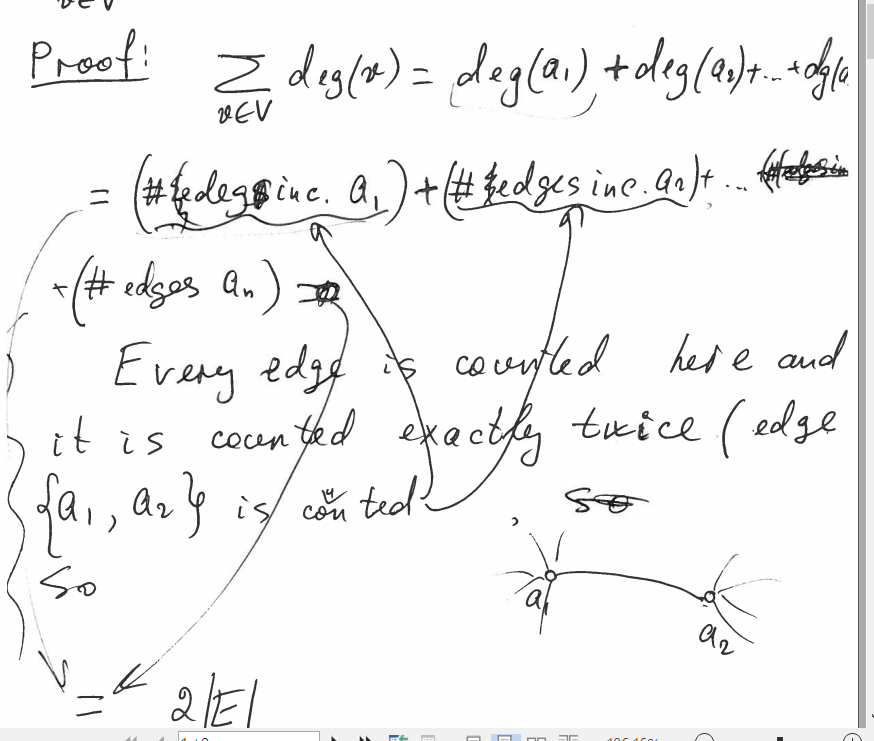
如果是incoming的， 叫做id (v)

如果是Outcoming的，叫做od(v)



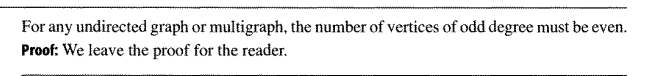
如果G是一个undirected graph或multigraph（multigraph就是两个点之间有多条路线），那么每个vertex的degree的和等于两倍edge

很容易证明，



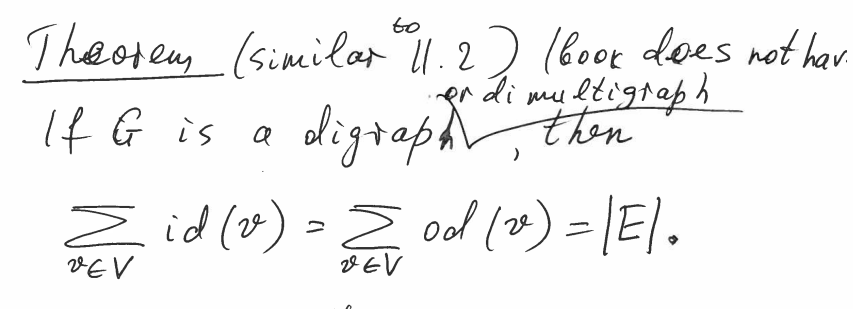
每个edge都被count了两遍

推出理论



拥有odd degree的vertices的数量一定是偶数，

理论：如果G是一个digraph，那么所有Indeg等于outdeg等于E



例子

Hypercube:对于一个n阶立方体，是一个有2^n个端点的undirected graph，

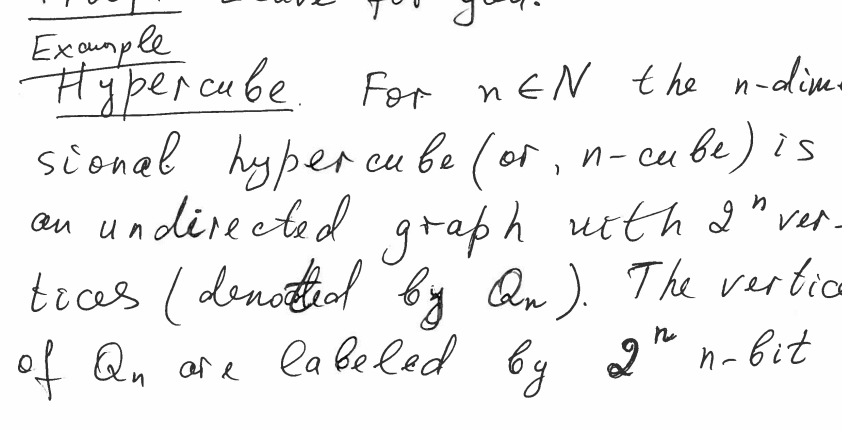
每一个端点都把他们从0,1,2,3,4,5,….2^n-1转换成对应的二进制

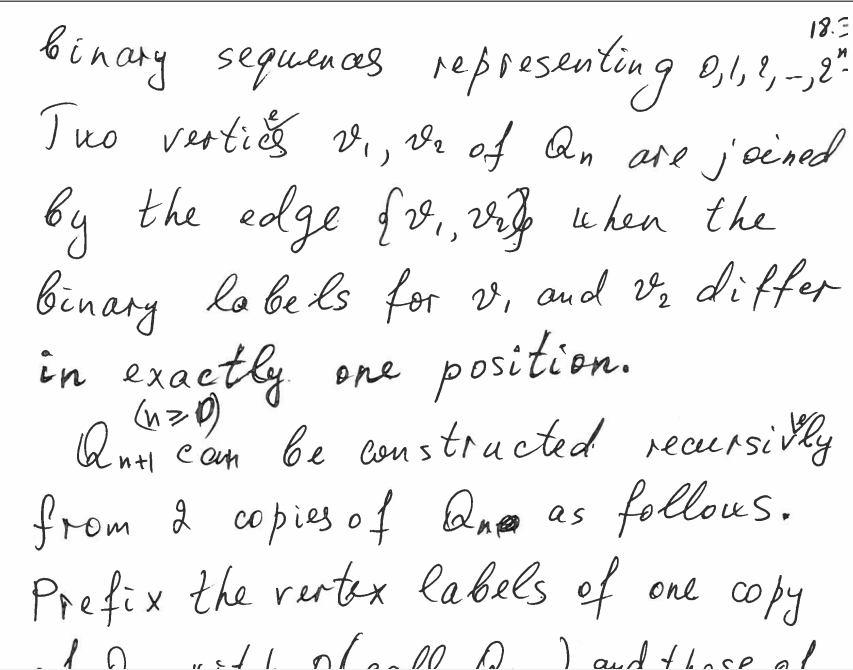
如果v1v2被 edge{v1,v2}所连，那么v1 v2的对应二进制必然只有一位不同

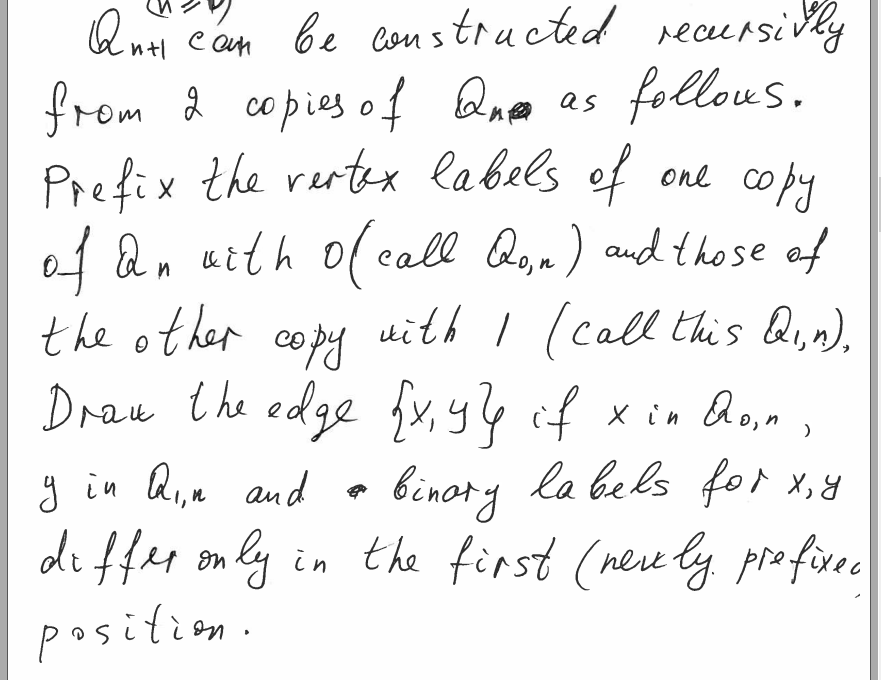
我们可以在这个QN基础上建造QN+1并不停循环，

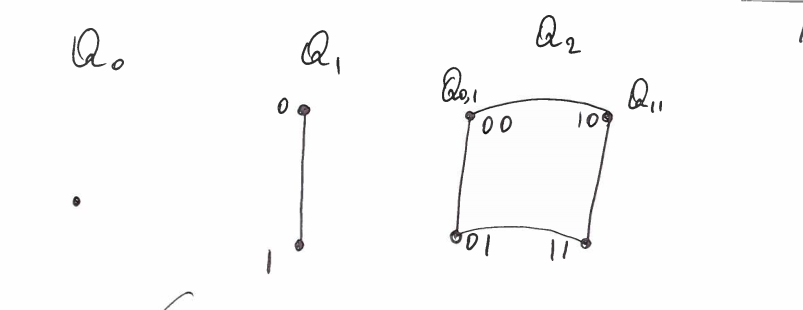
方法：我们复制两份QN，第一个叫做Q0N, 第二个叫做Q1N，

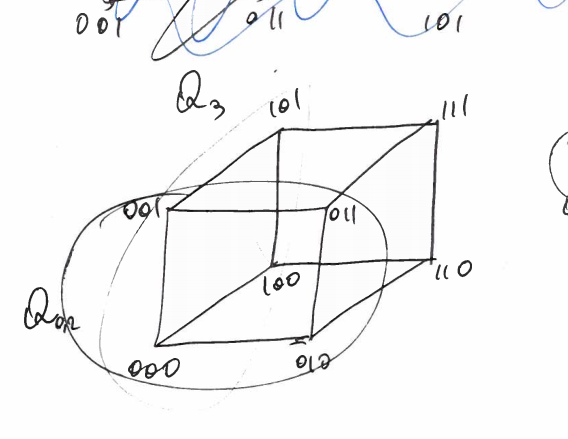
画出EDGE{X,Y},X属于Q0N,Y属于Q1N，那么我们只要让他们的首位不同就行了

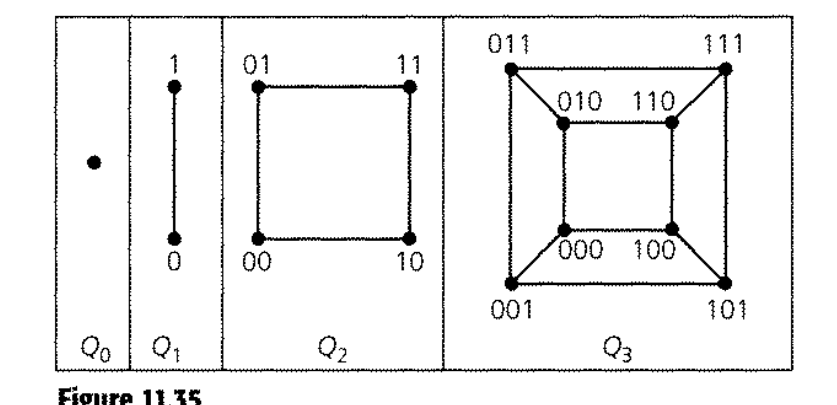
­­­­









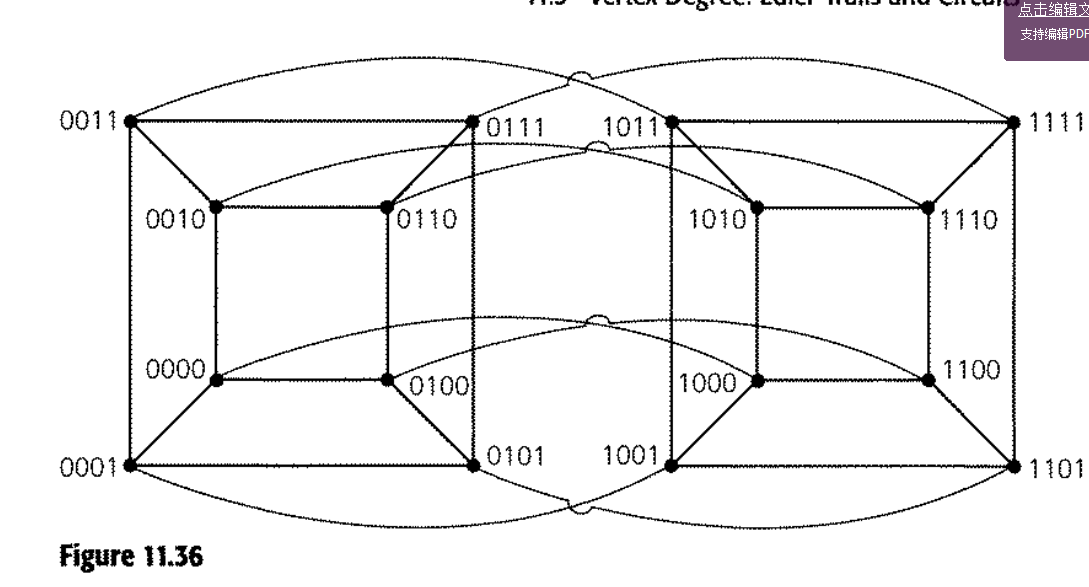


hyber cube的edge却不是单纯的✖2，因为每次都会多一部分将两者链接的点

计算方法就是



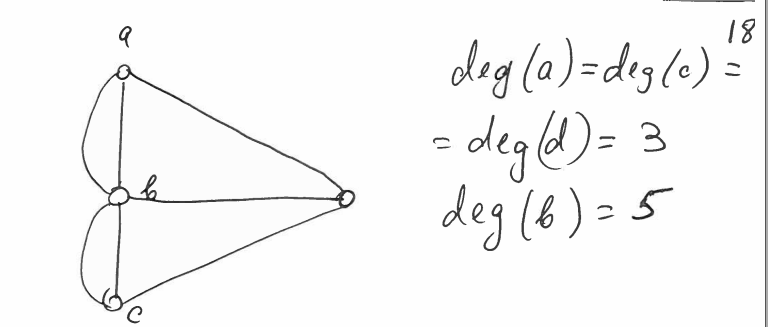
因为每一次翻倍，一个vertice的degree就会加一，因此是n，一共有2^n个点，然后根据公式，edge的数量等于所有点的degree的数量除以2

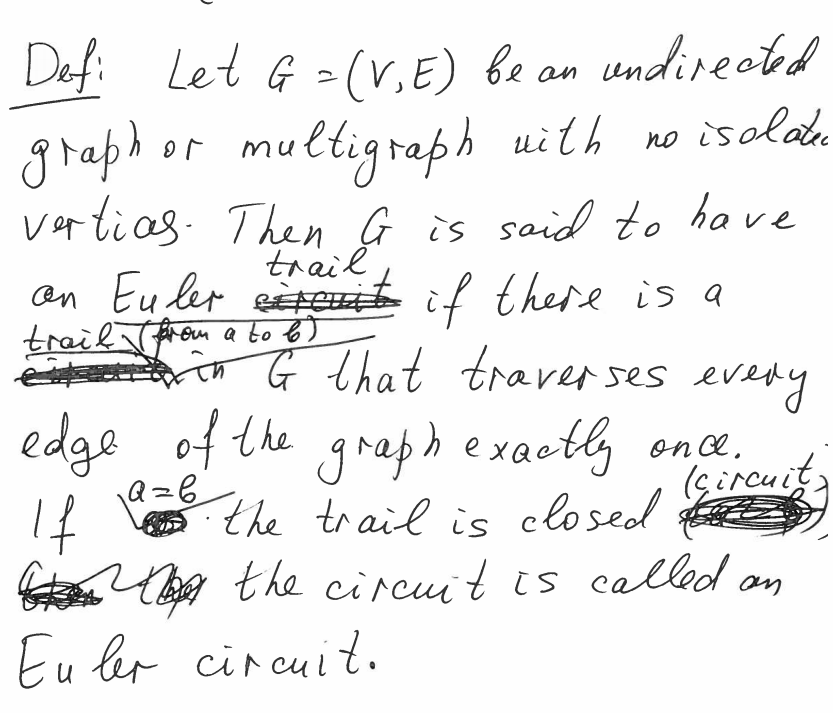


不要管pdf中那个网格例子， 他在说CPU怎样连接最快，网格例子都是失败例子

Seven bridges of ..

假设一个城市被分为四个点，七座桥，每个桥只走一次并返回原点的方式

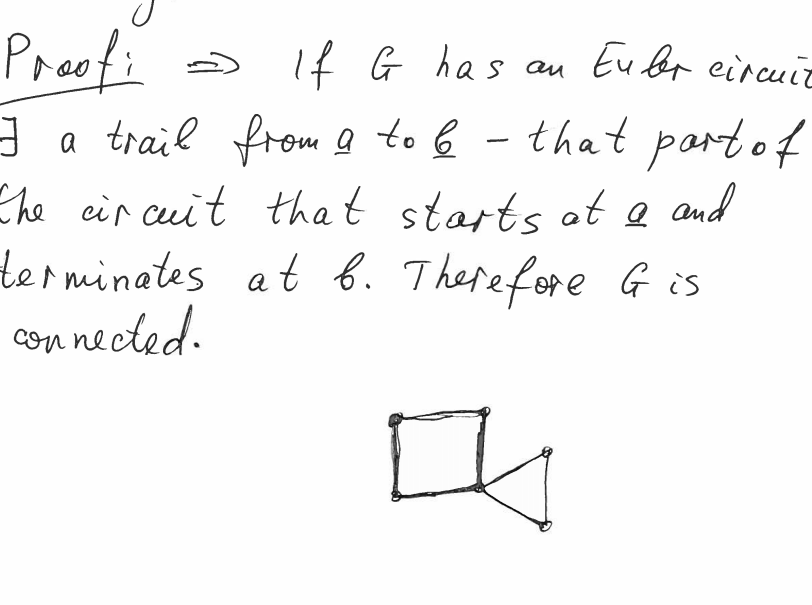




如果G是一个没有孤立点的undirected graph，那么如过从a到b如果有一个trail经过所有edge正好一次，我们就叫这个trail EULER TRAIL

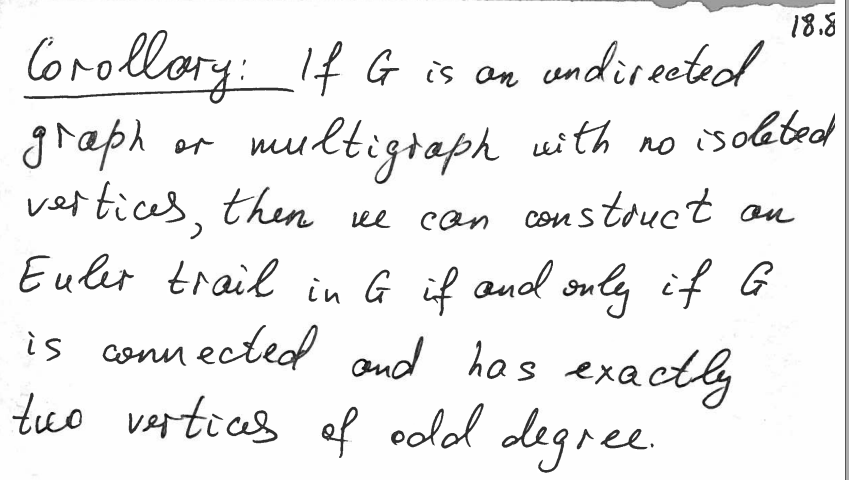
如果a=b，那么就是euler circuit

理论：G是一个没有孤立点的unddirected graph，那么G只有在connected并每个vertex都是even degree的时候才会有一个euler circuit

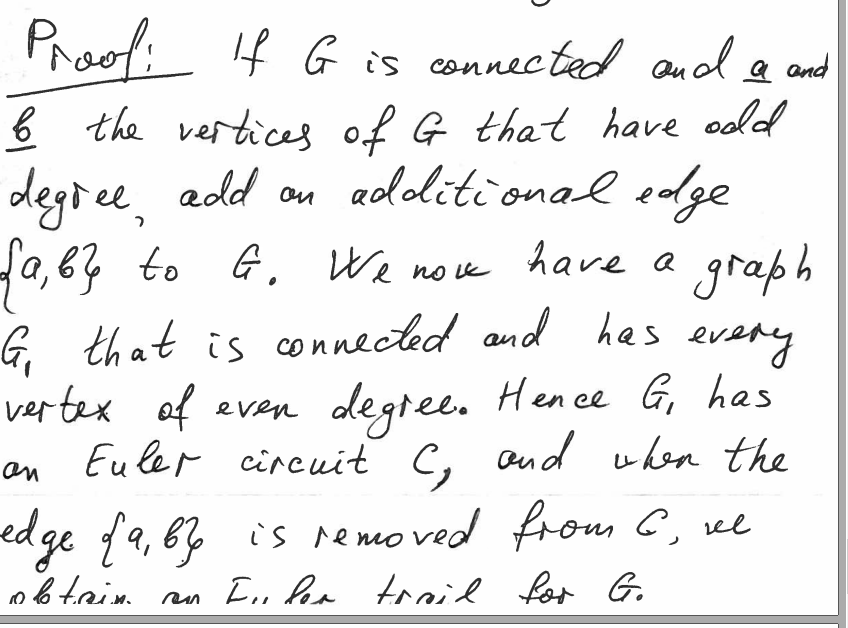


证明，如果有一个euler circuit,那么a到b必有路径，因此G就是connected的

Corollary推论：如果G是一个没有孤立点的undirected graph，我们可以建造一个euler trail当且仅当G是connected并且只有两个vertice是odd degree的时候

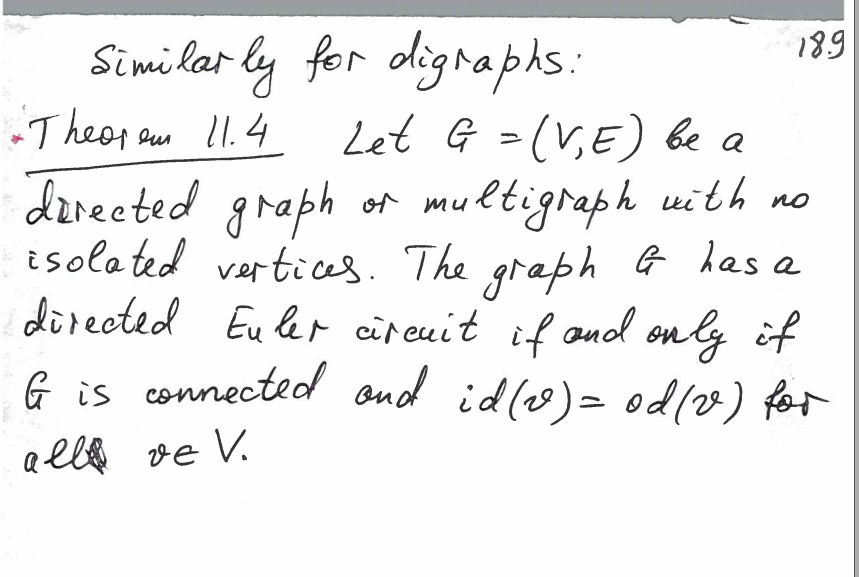


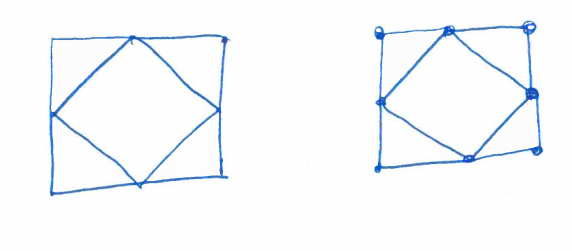
证明：如果G是connected,点a b为odd degree点，那么我们在两者之间加上一条edge，那么每个点都是even degree，那么就有了euler circuit。如果我们把这条edge移去，那么这就是a到b的euler trail

‘

Digraph同理，

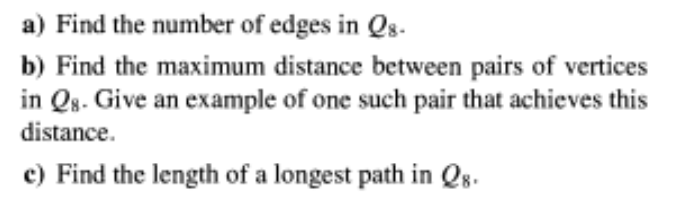
Graph只有G是connected且 每一个点的进degree=出degree id (v)=od(v)的时候才有Euler circuit





例题

8.



a/

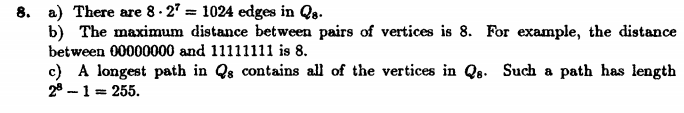
There are 2^8 vertices, the degree of every vertice is 8, and the edge = the sum of all degree/2

So the answer is 8\*2^8/2=8\*2^7

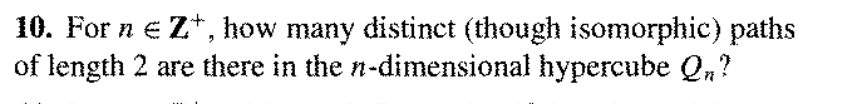
b/when every digit is completely different,like 00000000 to 11111111 , one diffrent digit means one distance .So the maximum distance is 8

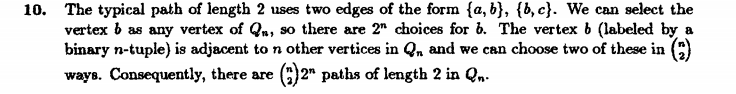
c/

It will cover all vertices, which is 2^8 -1



10



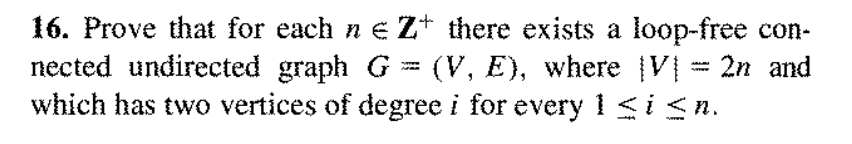


Firstly, there are 2^n vertices in Qn hybercube. assume it is vertex X

Then, for any vertex, there are n vertices adjacent it

So we can pick 2 vertices Y Z from n adjacent vertices, with the vertex X, we can build a path {Y,X}, {X,Z}

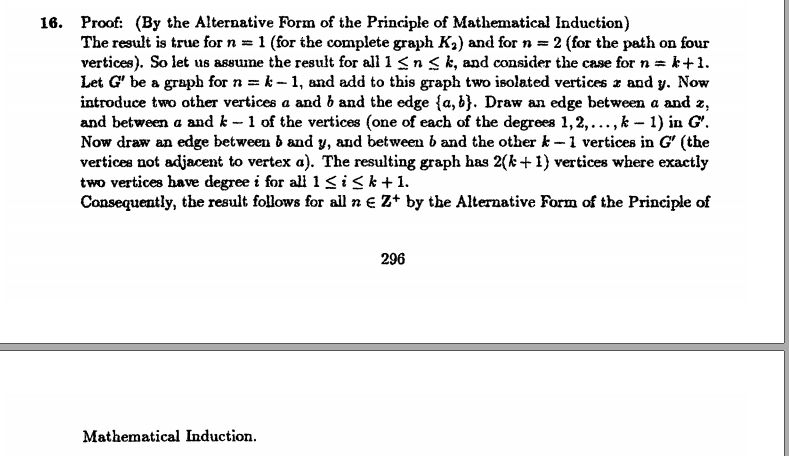
So there are nC2 \* 2^n paths.

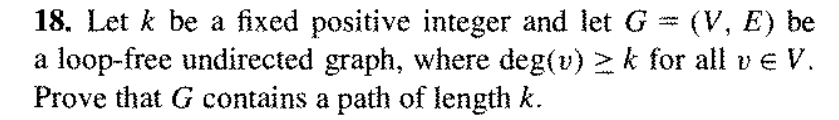


Step1: When n=1, we can build a path with 2 vertices, degree i=1,true

Step2: Assume for all k<=n, there are 2 vertices X, Y of degree i 1<=i<=k

Step3: Then for we can add 2 new vertices A, B to this graph, now we have 2(k+1) vertices which is , we connect vertice A to X, B to Y, now the graph is still connected, and the degree of X and Y will become 1<=i<=k+1 because the new vertices





Step 1: When k=1, cause deg>=1, there must be at least one path, true

Step 2: Assume for all m<=k, there is a graph with deg(v)>=m have a path of length m, and because the degree can >=m, there are at least m+1 vertices in this graph

Step 3: then we add a new vertice to the graph, and connect the new vertex to all old vertices, then deg(v) will >=m+1 for all old v, and the degree of new vertex will be the number of old vertices which is also >=m+1

And there will be an edge between the end vertex of length-m-path to the new vertex (cause it is connected to all vertice) , we add this edge to the length-m-path, and we will get a new path of length m+1.

