MATH 251 HOMEWORK ~~7~~ 8 SOLUTIONS

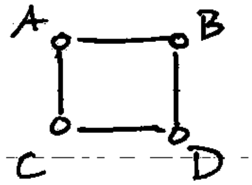
$\boxed{8.2.4}$ Minimum algorithm

Given n numbers $a_1, \dots, a_n$, find the minimum value $\alpha = \min\{a_1, \dots, a_n\}$.

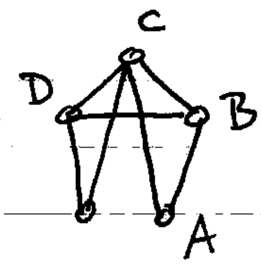
Step 1: Set $P = a_1$
Step 2: For $i = 2, \dots, n$
  If $a_i < P$, update $P = a_i$
Step 3: Output P.

Why does this work? Suppose a_j is the first number in our list equal to α. If $j=1$, then the condition of Step 2 is never satisfied, so the output is $P = a_1 = \alpha$.
If $j > 1$, then at the $(j-1)$st iteration of Step 2, we will satisfy the condition and set $P = a_j = \alpha$. Afterwards, the condition is never satisfied. Thus the output is $P = a_j = \alpha$.

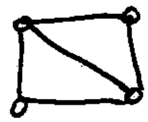
What is the complexity? For a list of n numbers we must make one comparison for each of the $n-1$ iterations of the loop in step 2. Thus the complexity of this for a list of n numbers is $f(n) = n - 1$.

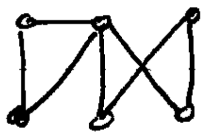
$\boxed{9.2.15}$ (a) $G_1 =$ 

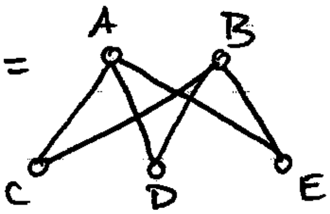
is a subgraph of

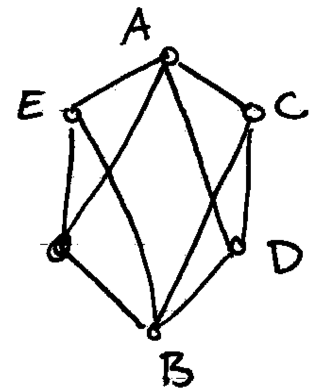
$G_2 =$ 

. One way is indicated by our labelling.

(b) $G_1 =$  is not a subgraph of

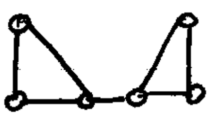
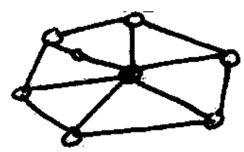
$G_2 =$  . G_2 only has one vertex with degree ≥ 3 but G_1 has two vertices with degree $= 3$.

(c) $G_1 =$ 

$G_2 =$ 

G_1 is a subgraph of G_2 labelling as indicated.

(remove the unlabelled vertex from G_2, the edges it touches and the edge CD.)

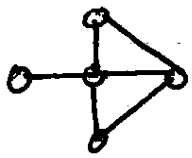
d) $G_1 =$  , $G_2 =$ 

G_1 is not a subgraph of G_2. In G_2 every pair of triangles shares a vertex.

9.2.18 The degree sequences a–c and e–g are impossible.

for (d), we have 

for (h), we have



Why are the others impossible?

(a) the sum of the given degrees is $4+4+4+3+2=17$ an odd number.

(b) the sum of the given degrees is
$100+99+98+\cdots+3+2+2+2 = 3+(1+2+\cdots+100)$
$= 553$, an odd number.

(c) There are to be 6 vertices, ~~two~~ two with degree 5. A vertex with degree 5 must be adjacent to all other vertices. Thus every vertex must have degree ≥ 2. The list contains a $\underline{1}$.

(e) There is a vertex of degree 5, so we need 6 vertices for a graph. There are only 5 degrees listed.

(f) This should have 6 vertices including one of degree 5, which must then be adjacent to all the others.
If we prune off the degree one vertices and their edges, we would construct a subgraph of degree list 4, 3, 3, 2. This is impossible by reasoning analogous to (e).

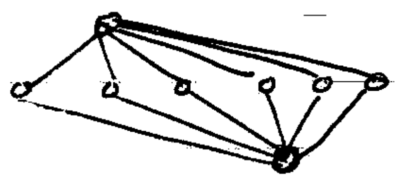
9.2.18 (cont.)

(g) There are to be 8 vertices including two of degree 6 and two of degree 1.

Let v_1, v_2 be the vertices of degree 6.

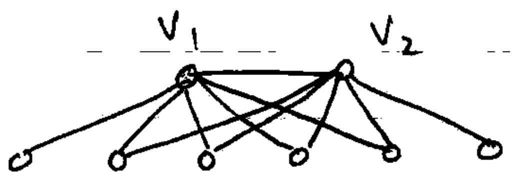
<u>Case 1</u> v_1 not adjacent to v_2.

Then we must have a subgraph isomorphic to $K_{2,6}$



this leaves no room for any degree 1 vertices.

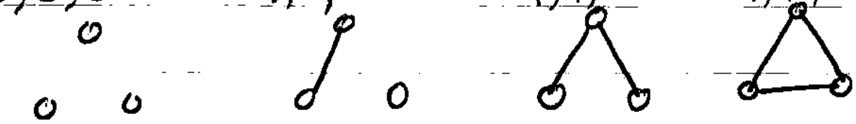
<u>Case 2</u> v_1 adjacent to v_2. ~~To have~~ ~~In this case, remove v_1, v_2~~ To have two degree 1 vertices, we must have that v_1, v_2 have exactly 4 common neighbors.



We can't increase just <u>one</u> vertex to degree 4 because adding an edge increases the degree of both vertices it touches.

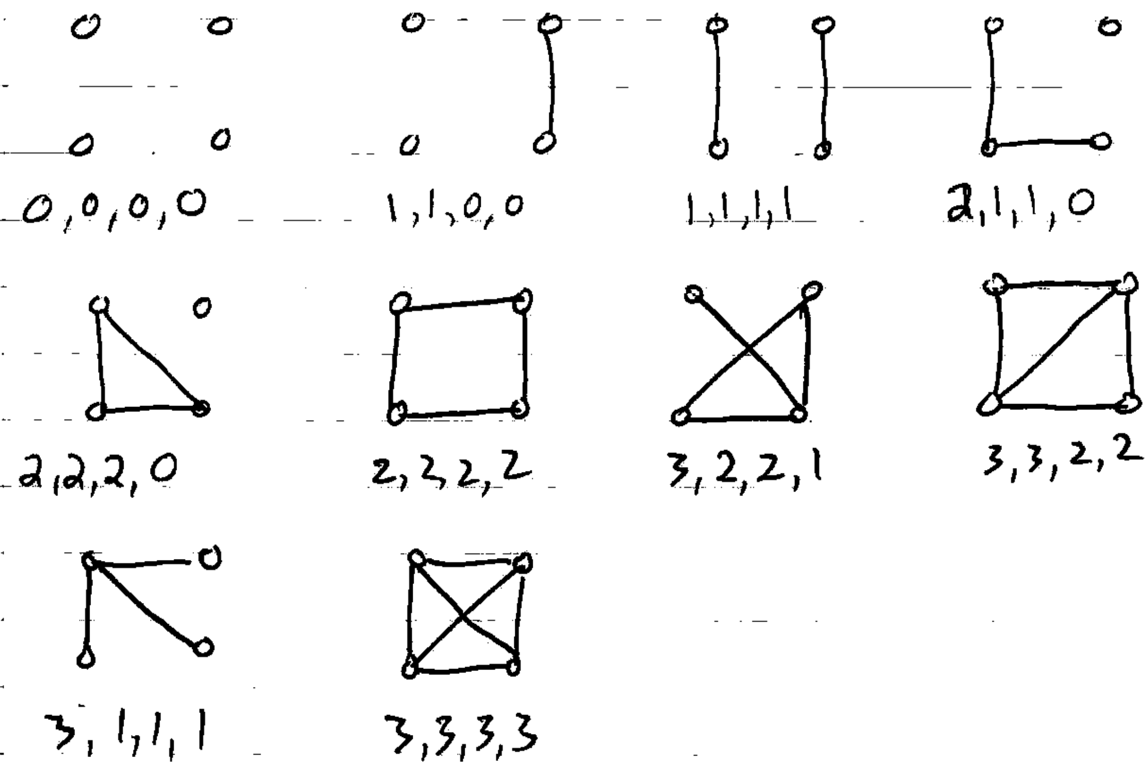
9.3.3 (a) All non-isomorphic graphs on three vertices.

0,0,0   1,1,0   1,1,0   1,1,1



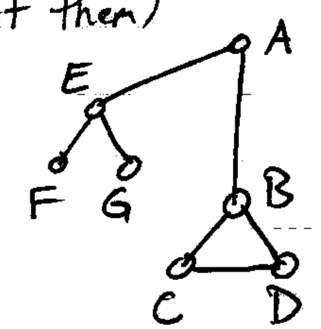
(i.e. find all subgraphs of the complete graph K_3 which have 3 vertices)

(b) All non-isomorphic graphs on four vertices



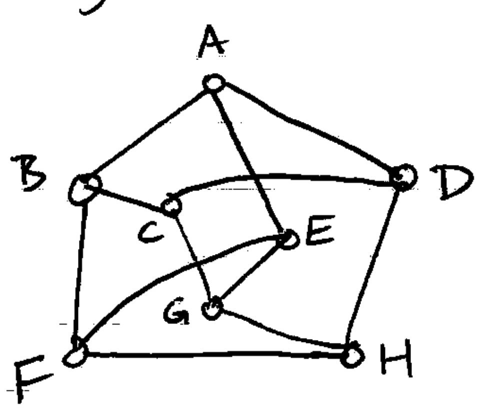
$\boxed{9.3.4}$ (a) These are not isomorphic. the graph on the right has a vertex of degree 4, the graph on the left does not.

(b) These are isomorphic. I'll relabel the graph on the right to show an isomorphism (there are four of them)



(c) These are not isomorphic, Each has a unique vertex of degree 3, B and r, respectively.
The vertices adjacent to B have degrees 2, 1, 1.
The vertices adjacent to r have degrees 2, 2, 1.

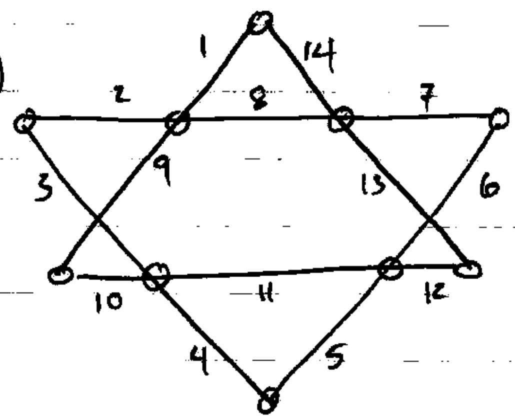
(d) These are isomorphic. One possible isomorphism is given by the following relabelling of the graph on the right



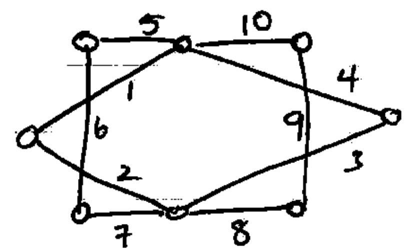
Remember: technically an isomorphism ~~from~~ is a bijection between vertex sets preserving edge relations...

$\boxed{10.1.4}$ (b) and (e) do not have Eulerian circuits.
(b) because it is not connected. (e) because it has vertices of odd degree.
For the others the labelled sequence of edges I found is:

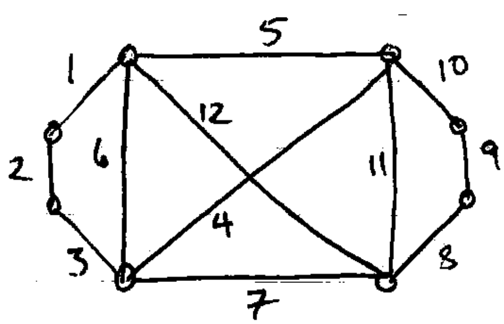
(a)



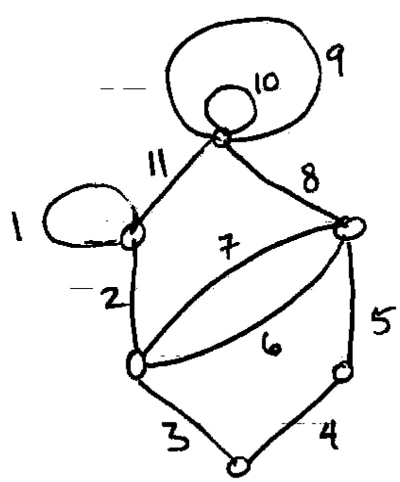
(c)



(d)

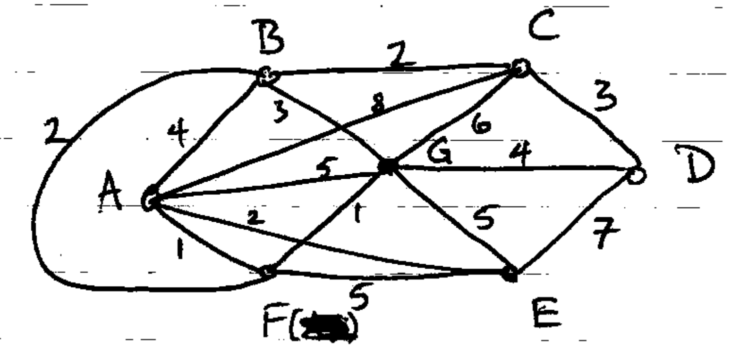


(f)



10.4.2

From A to E: just label 'em all!



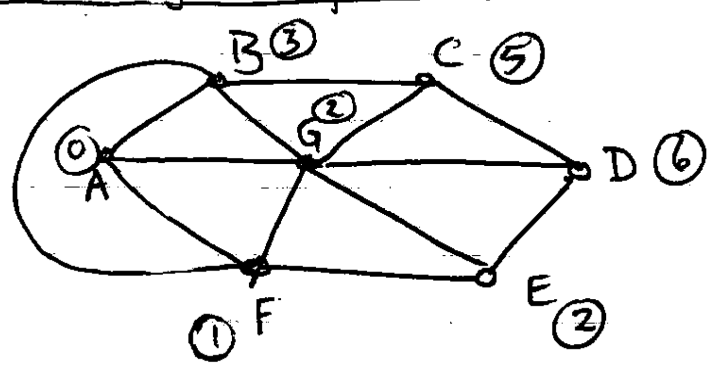
First Algorithm

1. A(–, 0)
2. B(A,4), C(A,8), E(A,2), G(A,5), F(A,1)
3. B(A,4), C(A,8), E(A,2), G(A,5), E(F,6), B(F,3)
4. B(A,4), C(A,8), G(A,5), E(F,6), B(F,3), G(F,2)
   G(E,7), D(E,9)
5. B(A,4), C(A,8), ~~G(A,5)~~, B(F,3), D(E,9), C(G,8)
   ~~C(A,8)~~ D(G,6)

6. C(A,8), C(B,5), C(G,8), D(G,6), D(E,9)
7. D(C,8), D(G,6), D(E,9)

$\longrightarrow$ all vertices are labelled!

Second algorithm produces



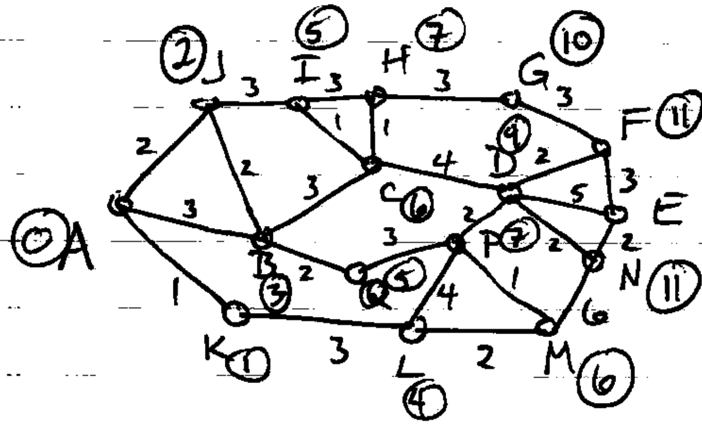
My process looked like this:

$\boxed{10.4.2}$ (cont.)

<u>Second Algorithm</u> (& cross out after an update)

1. A 0
2. ~~B4~~, ~~C8~~, ~~E2~~, ~~G5~~, ~~F1~~ → F 1
3. ~~B3~~, ~~G2~~, ~~E6~~ → E 2
4. ~~G7~~, ~~D9~~ → G 2
5. ~~B5~~, ~~C8~~, ~~D6~~, → ~~B3~~ B 3
6. ~~C5~~ → C 5
7. ~~D8~~ → D 6

$\boxed{10.4.10}$ Use Improved Dijkstra to find the shortest path from A to E. (I've added extra labels to aid my work!)



I'll box it when I keep it, cross it when I update it. (or don't need it)

keep label

1. A0
2. $\boxed{J2}$ $\boxed{B3}$ $\boxed{K1}$ → K1
3. $\boxed{L4}$ → J2
4. ~~B4~~, $\boxed{I5}$ → B3
5. $\boxed{C6}$, $\boxed{Q5}$ → L4
6. $\boxed{M6}$, ~~P8~~ → I5
7. ~~H8~~, ~~C6~~ → Q5
8. ~~P8~~ → C6
9. $\boxed{H7}$, ~~D10~~ → M6
10. ~~N12~~, $\boxed{P7}$ → H7
11. $\boxed{G10}$ → P7
12. $\boxed{D9}$ → D9
13. $\boxed{F11}$, ~~E14~~, $\boxed{N11}$ → G10
14. ~~F14~~ → F11
15. ~~E14~~ → N11
16. $\boxed{E13}$ → E13

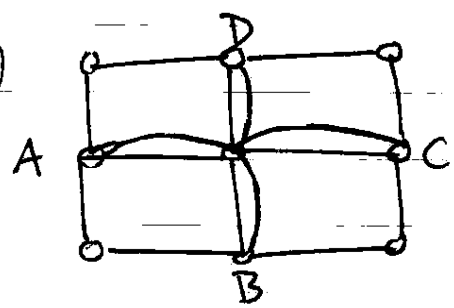
The shortest path has length 13 and is AKLMPDNE

(I traced back from E to A by moving up my list. The structure helps me remember when each permanent label gets assigned.)

10.4.12 The original algorithm can terminate before E is labelled if the graph is disconnected in such a way that there is no walk from A to E at all.

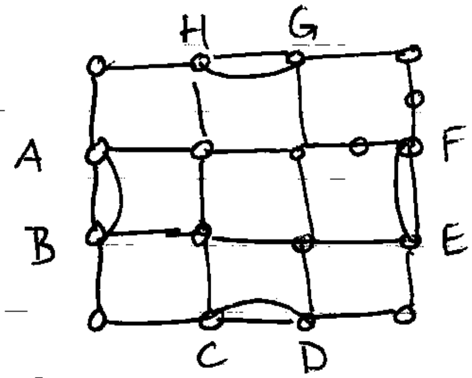
The same can happen in the improved algorithm. The temporary label ∞ on E ~~may~~ will never be updated if there is no path from A to E.

11.1.1 (a)



is one solution.

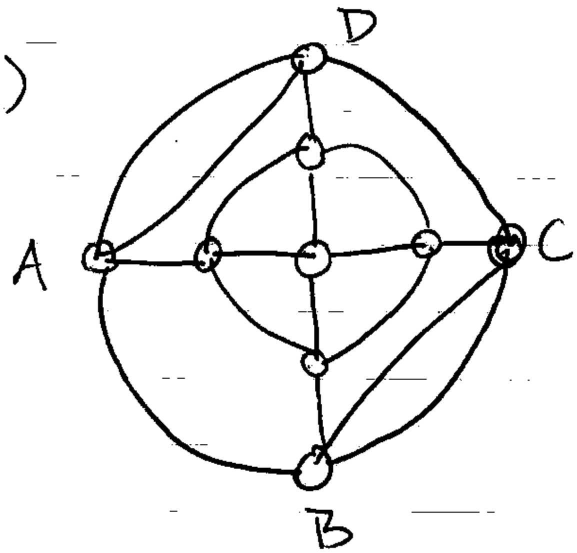
(b)



Odd vertices labelled.

Solution shown

(c)

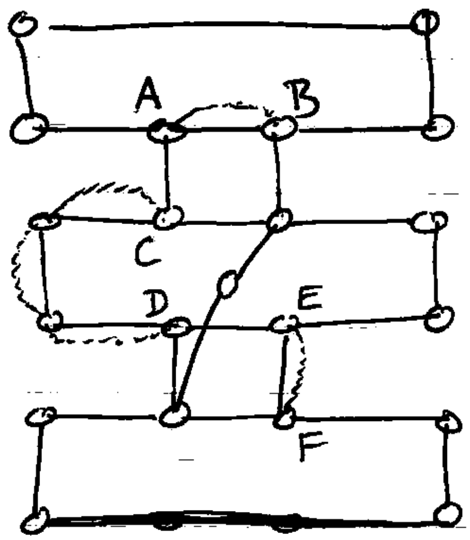


Odd vertices labelled

Sol'n shown

$\boxed{11.1.1}$ (cont.)

(d)



Odd vertices labelled.

6 of them
→ 15 possibilities.

AB CD EF has length $1+3+1=5$

so we see we never pair:

AD since $|AD|=4$, so partition ≥ 6

AE since $|AE|=5$, so " ≥ 7

AF since $|AF|=5$, so " ≥ 7

BD since $|BD|=4$, so ... ≥ 6

BE since $|BE|=4$, so ... ≥ 6

BF since $|BF|=4$, so ... ≥ 6

~~So we have two reasons~~

~~1) Pair AC~~

Now if we pair BC, we are forced into AD, AE or AF.

Therefore, we must have AB.

Cases not checked

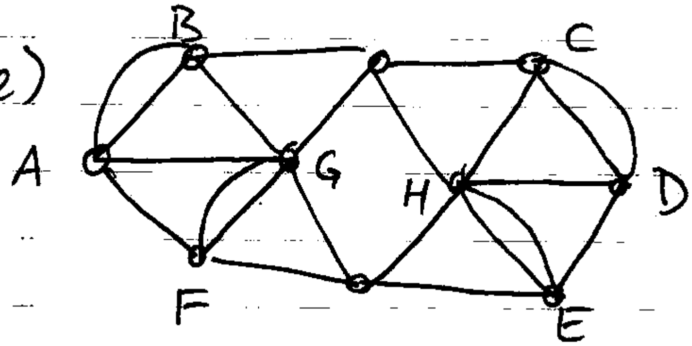
AB CE DF of length $1+4+2=7$

AB CF DE of length $1+4+1=6$.

So double the edges dashed above

$\boxed{11.1.1}$ (cont.)

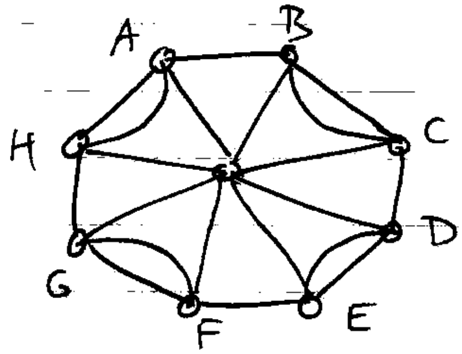
(e)



Odd vertices labelled

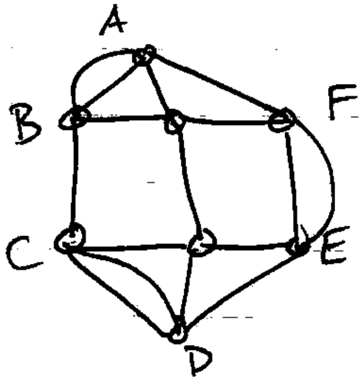
8 odd vertices, so we need at least 4 doubled edges. But this can be done by inspection as above.

(f)



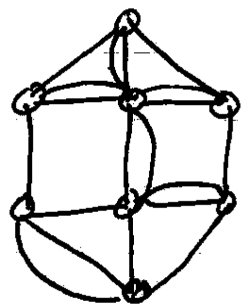
Odd vertices, again 8 of them, and by same argument as above we have soln at left.
(there are others)

~~[illegible]~~ $\boxed{11.1.5}$ (a)

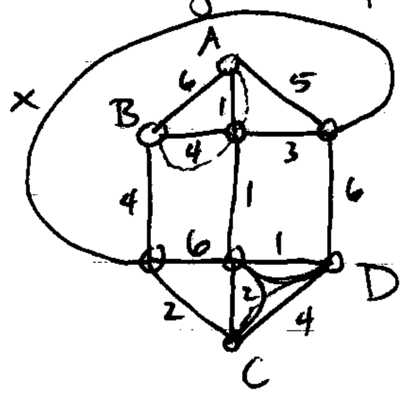


Odd vertices labelled 6 of them.
→ Need at least three doubled edges.
so soln at left.

(b) We did as the big class example.
soln is as below (I'll leave off all the labels)



(oops!)
(d) Like in above, but the added edge means there are only two pairs of odd vertices to work out

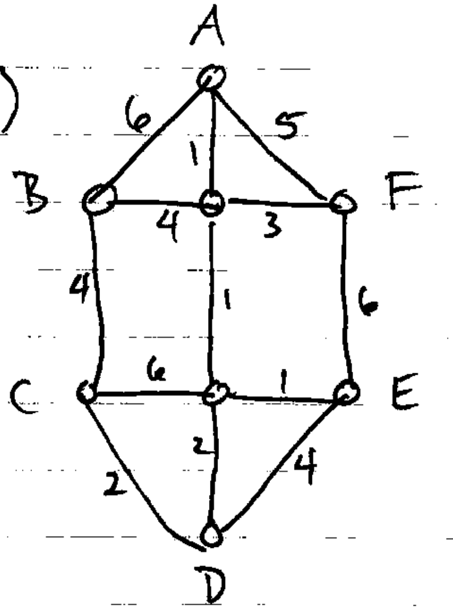


the book forgot to label the edge x but it doesn't matter. Even if $x=0$ we get the following

AB CD length $= 5+3=8$ *
AC BD length $= 4+6=10$
AD BC length $= 3+6=9$

Double the dashed edges!

(c)



Odd vertices labelled.
6 of them
→ 15 partitions to check.

No cleverness this time.
We'll just do it.
(Note only 15 paths to find!)

partition	length
AB CD EF	5+2+5 = 12
AB CE DF	5+5+6 = 16
AB CF DE	5+8+3 = 16
AC BD EF	6+6+5 = 17
AC BE DF	6+6+6 = 18
AC BF DE	6+7+3 = 16
AD BC EF	4+4+5 = 13
AD BE CF	4+6+8 = 18
AD BF CE	4+7+5 = 16
AE BC DF	3+4+6 = 13
AE BD CF	3+6+8 = 17
AE BF CD	3+7+2 = 12
AF BC DE	4+4+3 = 11
AF BD CE	4+6+5 = 15
AF BE CD	4+6+2 = 12

So we double edges along paths in the circled scenario: