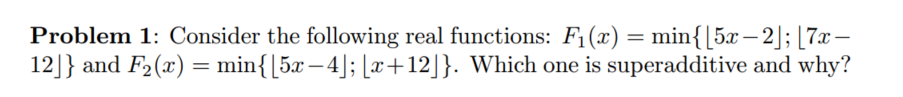
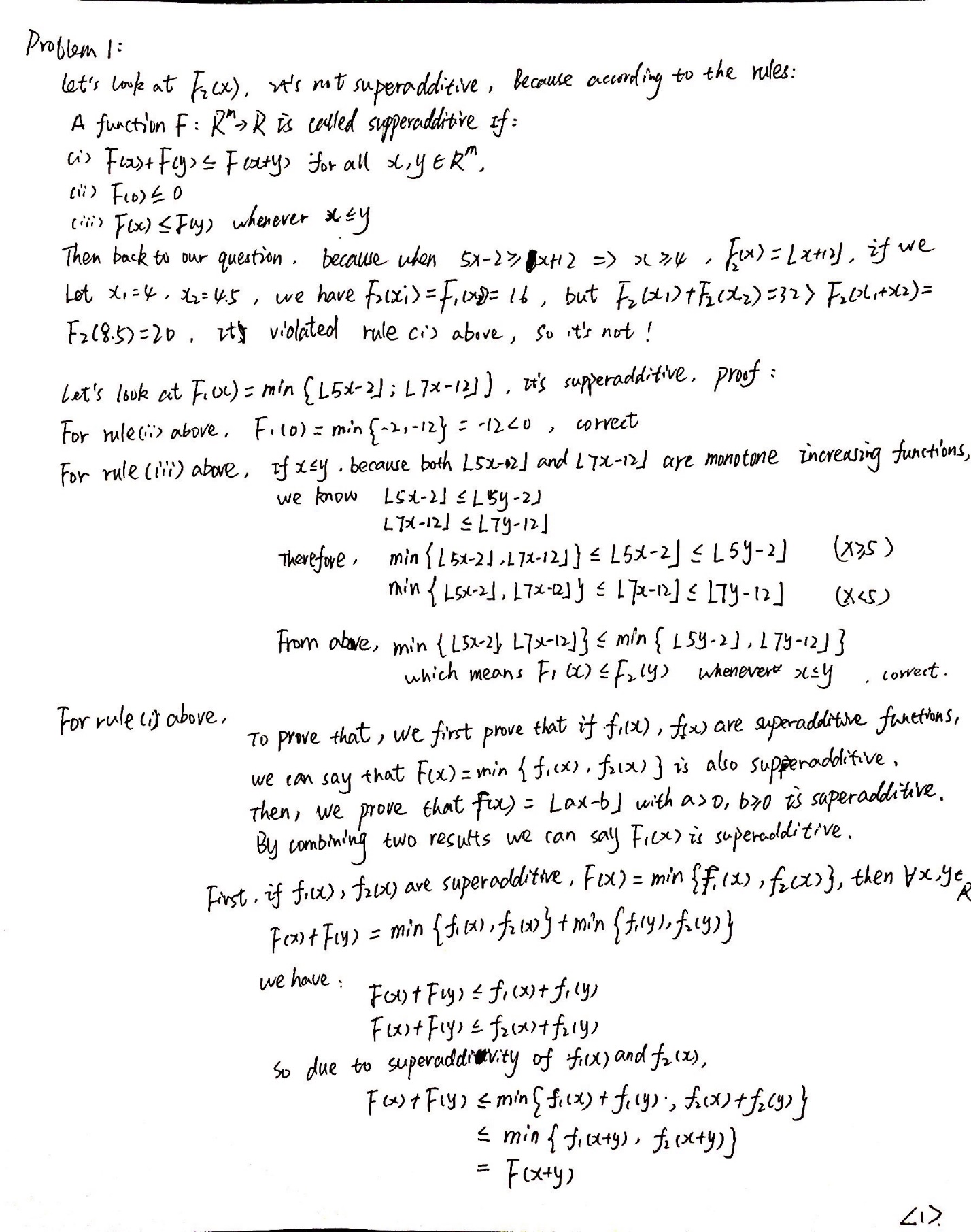
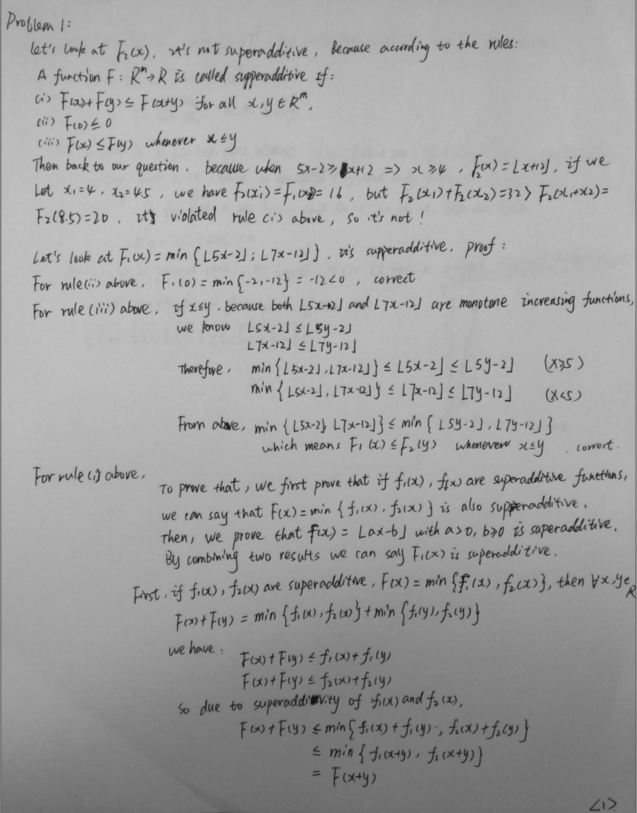
Name : Zhaofeng shang

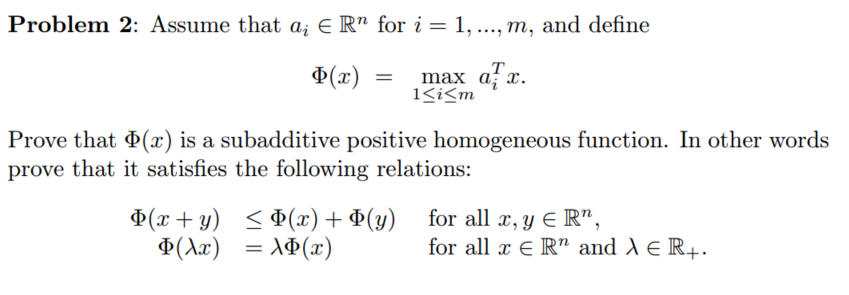
RUID : 181006463

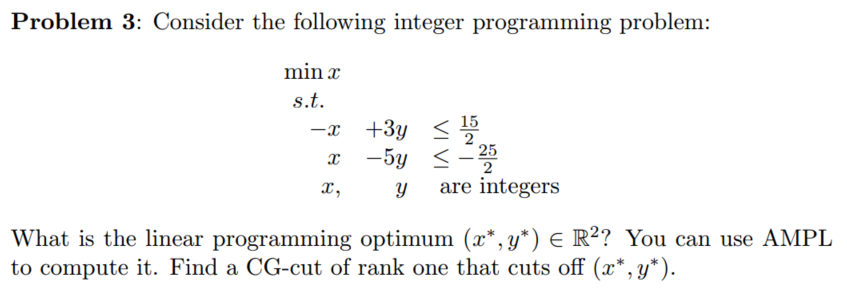


Ans:



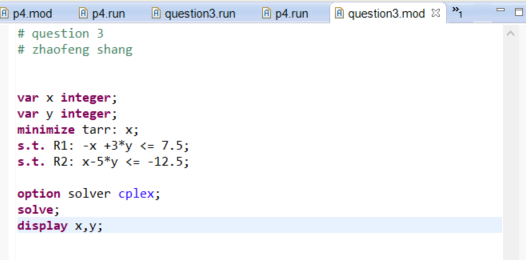




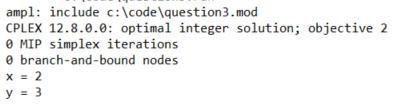


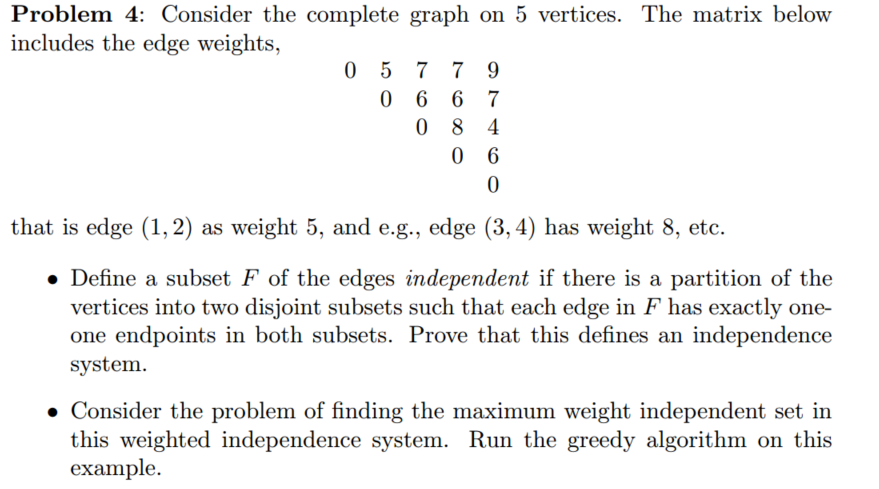
Ans:

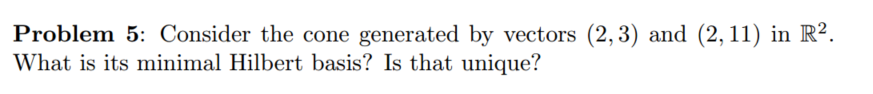
Code:



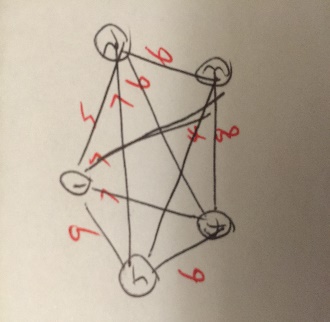
Result:







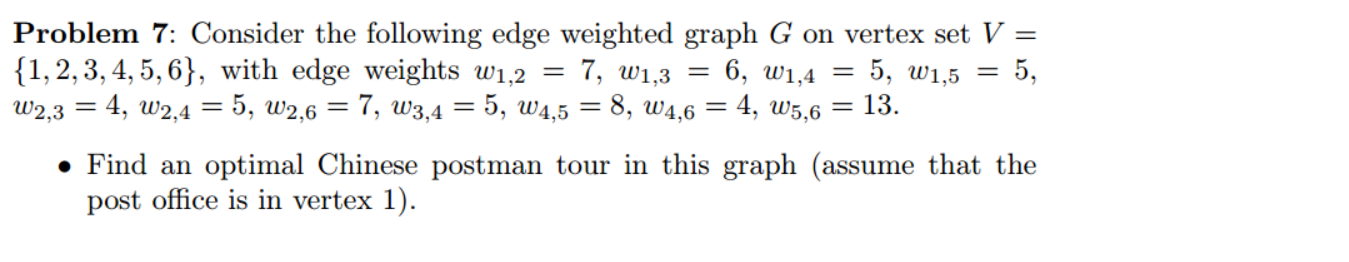
Ans:



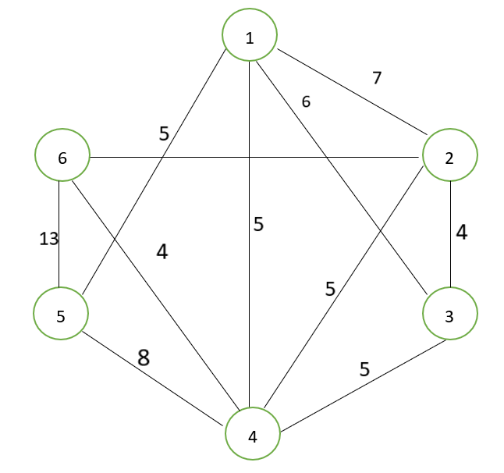
Q1: According to the theorem, a hypergraph (E, F), F ⊆ is called an independence system if

(M1) ∅ ∈ F;

(M2) If X ⊆ Y ∈ F, then X ∈ F.



Ans:



According to the question, the graph as above.

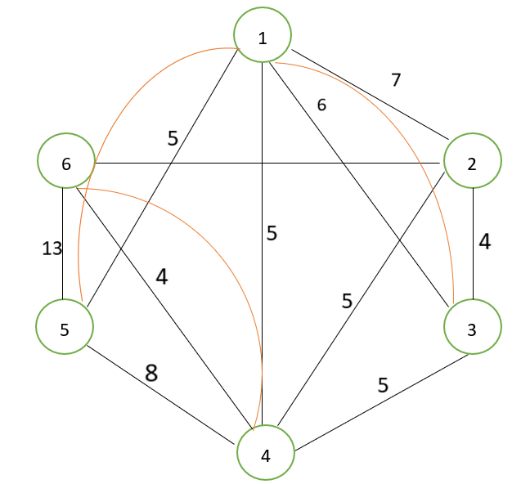
First, we need find all odd vertex, they are 3,4,5,6. Then we need do the matching.

Then, we need find all the connection pattern.

(3,4)&(5,6) cost 5+12=17

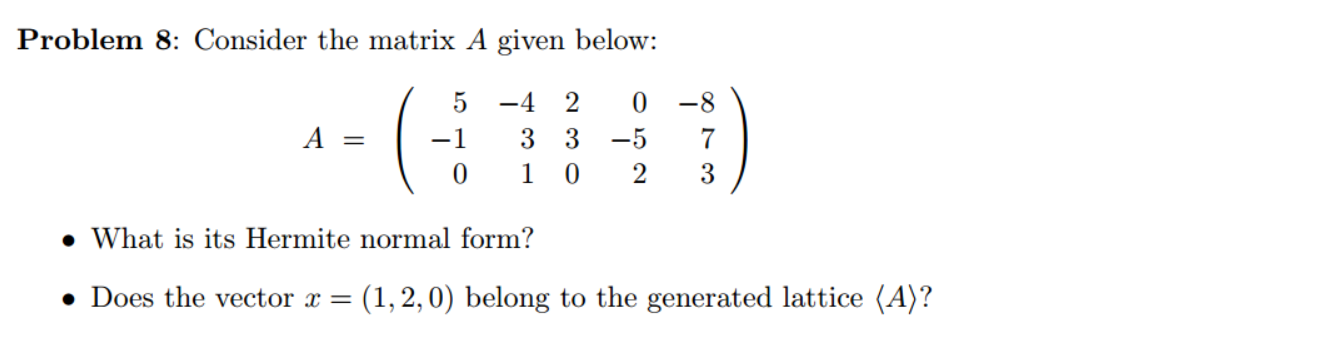
(3,5)&(4,6) cost 11+4=15

(3,6)&(4,5) cost 9+8=17

Because the second pattern has the lowest cost, So we choose this pattern to add the repeated arc, the final graph as below. 还要加上v26=7

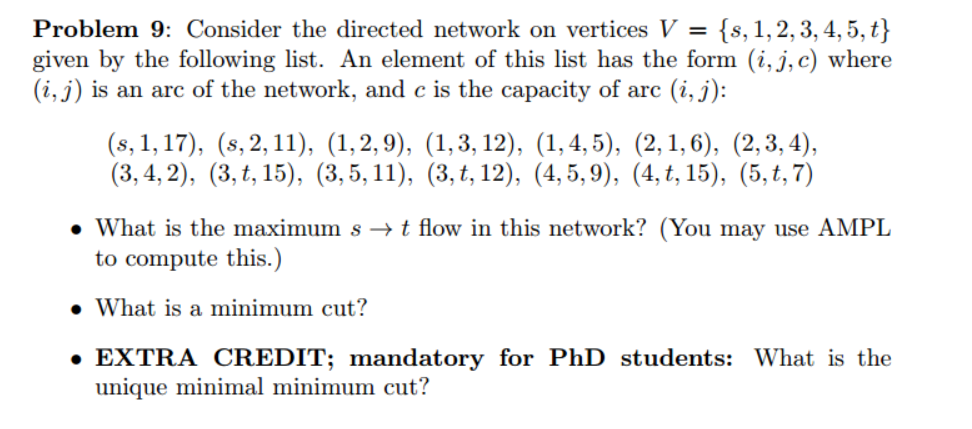
The tour is 1->2->3->1->3->4->2->6->5->4->6->4->1->5->1

The time he took is 84



https://www.mathworks.com/help/symbolic/mupad\_ref/linalg-hermiteform.html

https://www.youtube.com/watch?v=nypI4o7bxoo



<https://blog.csdn.net/mathsoperator/article/details/7215209>

1. option randseed 1;
2. param N >=2 integer;
3. param density >0,<1 :=20/N;
4. **set** cities:=1..N;
5. **set** links:=setof{i **in** 1..ceil(N\*N\*density)} (ceil(N\*Uniform01()),ceil(N\*Uniform01()));
6. param cap{links} := ceil(10+980\*Uniform01());
7. param s:=1;
8. param t:=N;
9. node nodes{i **in** cities};
10. arc F\_In>=0, to nodes[s];
11. arc F\_Out>=0, from nodes[t];
12. arc flow{(i,j) **in** links} >=0 <= cap[i,j]
13. from nodes[i] to nodes[j];
14. maximize mfv: F\_In;
15. data;
16. param N:=10000;
17. option solver cplex;
18. solve;
19. display mfv;

