CSCI 570 - HW 6

1.Proof:
SAT is defined by a set of Boolean clauses and variables
We need to reduce SAT it to Integer Programming instance with the same number of variables, let vi denote the constraints of each variable

0<=vi<=1

For every clause we will have a constraint that corresponds to it
This reduction can be done in polynomial time, it is hard to verify the satisfying assignment of integer programming than the given SAT instance
It is easier to verify that if the SAT instance has satisfying assignment then the corresponding integer programming instance has an integer solution

2.Proof:
Let TM be a turing machine which can reduce sat to half-sat with the given input F
F consists of variables x1,x2,…,xn
Construct a CNF consisting of variables x1,x2,…,xn and ~x1,~x2,…,~xn
Now construct a CNF consisting the clauses (xi v xj) ^ (~xi v ~xj)
If half-sat has a solution then sat will also have a solution
Example:
x1=T
x2=F
~x1=F
~x2=T
The CNF form will be as follows:
(x1 v x2) ^ (~x1 v ~x2)=T
Therefore sat has a solution if and only if half-sat also has a solution

3. Let C be the set of courses with starting and finishing time of each course
We can select at least k number of courses from the available courses, given no two courses overlap.
Now we can try to select the courses from the set which overlap, this can be done in polynomial time
We cannot overlap the courses, however we can choose k courses one after the other without overlapping if possible
So it is possible to choose verify the chosen k possible courses in polynomial time, however it is difficult to choose k courses in polynomial time, therefore this belongs to NP

4. while E′ != Ø do

let (u, v) let u and v be two vertices with different colours and and edge between them
VC ← VC U {u, v}
remove every edge in E′ incident on u or v

return VC

Let G=(V,E) be the planar graph with V vertices and E edges,
Let the minimum vertex cover of the graph G be VC = V/2
Let A be the solution produced by the algorithm and OP be the optimal solution

Approximation ratio >= A/OP
Therefore this problem is a 2-approximation problem

5. let G=(V,E) be the graph
T=(V,E0) be the resulting depth first tree
N represents the set of non leaf vertices and L be the set of leaf vertices

(i) Suppose there exists an edge e=(u,v) in E that is not covered by N, that is these vertices may be in L. Suppose the DFS has explored the vertex u and then it will search for next available vertex which is not explored, so in this case it will explore the vertex v from vertex u along the edge e. Since vertex v is explored after vertex u, this means vertex u belongs to the set N (non leaf), therefore the initial assumption contradicts and therefore convers every edge in E

(ii) For this we need to create a matching in G
For every non leaf vertex u, pick an edge eu to one of the next vertex. Let OD be the vertices on the odd depth level and EV be the set of vertices on the even depth level, compare the two sets and assign Big:=OD if odd set is greater than even set or Big:=EV if even set is more than the odd set. Let the Big be atleast N/2, since non leaf vertices are N. Let the edge set M be {uEe|u belongs to Big}, and since M is matching the optimal vertex cover, then it has to contain at least N/2 vertices when there N vertices, so the solution is at worst a 2-approximation