Homework 4

1. Problem 34-1 in the textbook.
   a. $INDEPENDENTSET = \{< G, k > | Graph\ G\ has\ an\ idependent\ set\ of\ size\ k\}$.
   Given a graph G and an integer k, is there an independent set in G of size k.
   Prove independent set problem is NP-complete.
      1) First, prove that independent set problem is in NP. For a given graph $G =< V, E >$ and an integer k. we use the set $V' \subseteq V$ of vertices in the independent set as a certificate for G. The verification algorithm affirms that $|V'| = k$, and then it checks that there is no edge between any pair of vertices in V'. This verification can be performed in polynomial time. Thus, $IS \in NP$.
      2) Then, prove that independent set problem is in NP-hard by showing that $CLIQUE \leq_p IS$. Given a graph G and an integer k, compute the complement graph $\bar{G}$. To compute the complement graph $\bar{G}$, we connect two vertices $\bar{G}$ which are not connected in G. That is $\bar{G} = (V,\ \bar{E})$, $\bar{E} = \{(u, v)|(u, v) \notin E$. The transformation is apparently polynomial time. It is clear that if G has a clique $V' \subseteq V$ with $|V'| = k$, then $\bar{G}$ has an independent set which corresponds to that clique in G, verse vice. Thus, $IS \in NP - hard$.
      3) Since $IS \in NP$ and $IS \in NP - hard$, independent set problem is NP-complete.
   b. Given the black box IS (G,k) to solve IS problem. We can use the following two steps to find the maximum IS.
      1) Find k

   MAX-IS

```
for k←|V| down to 1
    do if G has an independent set V′ of size k
           then return k and V′
```

   Since the queries to the black box are counted as a single step, the running time is $O(|V|)$.

   c. Obtain the vertex set.

   Iset = null, i = K

   While (i>0)

      Select a vertex v in G, G'=G

      For each vertex u in G', such that u is not adjacent to v.

         Add an edge between v and u in G'.

      If IS(G', i) = false

         Add v to Iset. Remove v and any vertex adjacent to v from G.

Homework 4

i--

return Iset

the running time is $O(|V|^2)$.

d. When each vertex in graph G has degree 2, the graph is a cycle. We start at any vertex in the graph, and pick each alternative vertex in the cycle until the size of independent set is $\lfloor |V|/2 \rfloor$.
The running time of this algorithm is $O(|V|)$.
e. If ~~G is bipartite, the independent set of maximum size is the side with larger number of vertices. The running time of this algorithm is~~ ~~$O(|V|)$.~~

2. Problem 34-2 in the textbook.
a. It is polynomial time solvable.
Assume that there are m coins worth x dollars, so there are $(n-m)$ coins worth y dollars. The total amount is $S = mx + (n-m)y$. Each of them wants to take $S/2$ dollars.
Suppose Bonnie take a coins worth x dollars, $0 \leq a \leq m$. We check that how many coins worth y dollars Bonnie can get. Obviously, check if $ax \leq S/2$ and $\left\lfloor \left(\frac{S}{2} - ax\right) / y \right\rfloor \leq (n-m)$. Bonnie can take $0, 1, \ldots, m$ coins with x, so there are at most m+1 possibility to check.
b. It is polynomial time solvable.
Sort all the coins by decreasing order. First give Bonnie the coin with the largest denomination d_m, then give Clyde coins from the remaining coins until he gets the same amount as Bonnie. After giving the coin with d_m to Bonnie, if the sum of the remaining coins are not less than d_m, Clyde will get the same amount as Bonnie. After that, repeat the previous process until all coins are gone.
If after giving Bonnie a coin with denomination d, the sum of the remaining coins is less than d, the money cannot be divided exactly evenly.
c. NP-complete.
1) Given a bag of checks and a division of the checks, we simply sum Bonnie's checks and Clyde's checks, and verify whether the sum is same. These are obviously accomplished in polynomial time. Thus, this problem is in NP.
2) We prove that Bonnie and Clyde checks division (BCCD) problem is in NP-hard by showing that $SUBSET-SUM \leq_p BCCD$. The $SUBSET-SUM$ is that given a set S of numbers, is there a subset $S' \subseteq S$ such that the sum of all numbers in S' is exactly t. Let T be the sum of all numbers of set S, we add a new number $(T-2t)$ to S. Each number in S corresponds to a check. Let the new set be U, if there exists a subset $S' \subseteq S$ such that the sum of all numbers in S' is exactly t, U can be divided into two parts, each of which has the same sum $(T-t)$. If there exists a pair of partitions of U, such that the sum in each

partition is $(T-t)$ and one of the partitions contains the number $(T-2t)$, we remove this number from U. Then, we get a set, which contains a subset whose sum of all numbers equals t. Thus BCCD problem is in NP-hard.

3) The Bonnie and Clyde checks division problem is NP-complete.

d. NP-complete.
It is apparently in NP. The proof is similar to that in part (c). To prove this problem is in NP-hard, we reduce BCCD problem in part (c). In part (c), we have a set of checks C, and decide whether it can be divided evenly. To construct a set of checks in part (d), we multiply every element in C by a number greater than 1000, say m, and call this new set C'. If C can be divided evenly, C' has a solution whose difference is not larger than 100. If C' has a division in which the difference is no larger than 100, than C has an even division. Because every element in C is corresponding element in C' divided by m, the difference is so small that we can ignore it. Thus part(c) has an even division. Therefore, this problem is NP-complete.

3. Problem 35-1 in the textbook.
   a. We reduce $SUBSET-SUM$ to $Bin-Packing$. The $SUBSET-SUM$ is that given a set S of numbers, is there a subset $S' \subseteq S$ such that the sum of all numbers in S' is exactly t. Let T be the sum of all numbers of set S, we consider a case of $t = T/2$. That is $< S, \frac{T}{2} >$. Then we construct the following $Bin-Packing$ instance, by dividing each element in S by $T/2$. The instance of $Bin-Packing < C, 2 >$, whether elements in set C can be fit in two bins with size 1. If there exists a subset $S' \subseteq S$ such that the sum of all numbers in S' is exactly $\frac{T}{2}$, S''s corresponding elements in C can be fit in one bin, and the remaining elements in $S - S'$ are fit in the other bin.
   b. The total number of bins required must be greater than or equal to S. Since the bins in this problem are unit-sized, at least $\lceil S \rceil$ bins are required.
   c. We prove this by contradiction. Assume 2 bins are less than half full, the first one is b_1, and the second one is b_2. The size of the object O fit in b_2 is less than $1/2$. According to the $First-Fit$ heuristic, O is placed b_2, since b_2 is the first bin than can hold it. However, O is less than $1/2$, which means it is fit in b_1 with more than $1/2$ capacity. It contradicts to the assumption. Thus, at most one bin is less than half full.
   d. In the worse case, every bin is a little greater than half full. Thus, at most $\lceil 2S \rceil$ bins are required.
   e. Based on part (b) and part (d), $C^* \geq \lceil S \rceil$ and $C \leq \lceil 2S \rceil$. Thus, $\frac{C}{C^*} \leq 2$.
   f. First-Fit(S)

```
1   Sort elements in S by increasing order and put them in an array A[1,…,n]
2   current_capacity=1
3   j←1                    ⊳j is the number of bins required
4   for i←1 to length(A)
```

Homework 4

```
do if A[i]≤current_capacity
      then current_capacity←current_capacity-A[i]
              i++
      else current_capacity←1
              j++
return j
```

34-2 (e)

For any graph G with |V|=n, let us suppose that the size of maximal independent set IS is x, and the size of minimal vertex cover C (34.5.2) is y, we prove that **x+y = n.**

First, for any independent set IS', V-IS' is a vertex cover. If there is an edge e_{v1v2} which cannot be covered by V-IS', then both v1 and v2 belong to IS', which is contradict to the fact that any two vertices in IS' are not adjacent to each other. Similarly, we can prove that for any vertex cover C', V-C' is an independent set.

If IS is a maximal independent set, then V-IS is a minimal vertex cover. Thus, x+y = n.

For any bipartite graph G with left vertex set L and right vertex set R, let us suppose that the size of the maximal matching M is m, the size of minimal vertex cover C is y, we prove that **m=y.** We add a source s on the left side and a sink t on the right side as in figure 26.8 (b).

Let A be a subset of L and B a subset of R. We consider the set K = A U B. Let S = {s}U (L-A) U B and T = {t} U (R-B) U A. Next we prove that K is a vertex cover of the graph if and only if (S,T) is a cut of the network with finite capacity.

Suppose (S,T) is a cut of finite capacity. Then no arcs from L to R are in this cut, in particular there are none from L-A to R-B in the cut. So, all arcs from A go to R-B and all arcs from L-A go to B. Therefore, every edge of the graph has one vertex in A or one vertex in B, which indicates that K is a cover. Note that the arcs in the cut that start froms all go to A, and those that end at t all start in B, each having capacity 1, so the capacity of the cut is |A U B|.

Suppose K is a vertex cover of the graph. Let A be the intersection of K with X and B the intersection of K with Y. This forms S and T as before. Consider an arc from a vertex in S to one in T. If this arc starts from s, it must go to A (and so, has capacity 1). If it starts in X-A, it must go to B since K is a vertex cover. So this arc does not go to T. If it starts in B, then it must go to t, with capacity 1. Thus, all arcs from S to T have finite capacity, so the cut (S, T) has finite capacity.

Next we prove that the size of a maximum matching is equal to the size of a minimum vertex cover (m=y).

Consider the corresponding network of the bipartite graph. The value of this max flow is the number of edges in a maximum matching. By the Ford-Fulkerson theorem, this is the minimum capacity of any (S, T). Thus, m=y.

Now we have m=y, x+y = n. So we have x = n-m. Formally, |IS| = |V|-|M|, the size of maximal independent set is equal to the number of vertices minus the size of the maximal matching.