1.

2. To see ~~bigest~~ largest O(sqrt(n)) number
   k <= O(sqrt(n))
   Total time = kth samllest + sort's
   = O(n) + O(sqrt(n) * log(sqrt(n))) = O(n)

3. a) optimal value for each and every node has j containing >1 where the node contain i.

Opt(i) = max(Opt(j)+1)

Ove[i] = longest path between V1 to Vi

Ove[i] . Max(ove[i]+1)+1 while v<1

```
int LongestPath() {
  for (int i=1; i<=V; --i) {
    ove[i] = -1;
    Ove[i] = 0;
      for(int i=1; i<=V, i++){
      if(ove[i] >=0) {
          for( i in 'v' to V) {
          Ove[v] = Max(ove[v]+1, ove[v]);]]
            return ove[v];}
```

b) - Two nodes can be connected such thee V1 is Connected to (V2, V3 .. Vn)

- Vj is connected with (V(i+1), (i+2), .. Vn)
- number of vertices as V=n

edges = E = n x (n+1)/2 = O(V²)

Opti[i] = Max(1<=k<i) ∈ E(Opti[k]+1)

- P[1]=0
- for i=2 to n
- Algorithm see for all edges going into i store all max longest Path has edge +1
- It set P[i] to this value
- it the last complete P[n]

Runtime = O(n²)

4. One bowl = 40 = 1 h and 4 lbs
one mug = 50 = 2 h and 3 lbs
Total = 15

	Bowls	Mugs
No #	24	8
Profit	40	50
Color	1	2
clay	4	3

~~Maximize Profit~~ Maximum Profit is 136.

5. To find longest Palindromic subsequence, we have find string that given and reverse the given string.

```
int dp[1000][1000];
int lcs(string s, string t, int n, int m){
  if(n==0||m==0) return 0;
  if(dp[n][m]!=-1) return dp[n][m];
  if(s[n-1]==t[m-1]) return dp[n][m]=1+lcs
    (s,t,n-1,m-1);
  else return dp[n][m]=max(lcs(s,t,n-1,m),
    lcs(s,t,n,m-1));
int main(){
  int n; cin>>n
  string s; cin>>s;
  string t=s;
  reverse(t.begin(), t.end());
  memset(dp,-1,sizeof(dp));
  cout<<lcs(t,s,n,n);
}
```

4. Part 1) - Every problem in NP can be reduced to the said problem in polynomial time
- It is in NP.

Part 2) The maximum ~~in~~independent set Problem of a graph can be reduced to the largest complete subgraph

1. Find all the independent vertices in the graph.
2. Take set C be the complement graph of MIS. It has the vertices that are not in MIS and has edges that are not ~~incl~~ incident of any of the vertices in MIS.
3. If number of vertices in C is $\geq k$, answer is yes, else no.

Complete subgraph of size = n - |MIS|.