201810986 임재훈

1. a). $1+(4\times3)+2+(3\times4)+2 = 29$

b) $OPT[i] = \begin{cases} Max(OPT[i-1]+v_i,\ OPT[i-2]+v_i \times v_{i-1}) & (i\geq2) \\ v_1 & (i=1) \\ 0 & (i=0) \end{cases}$

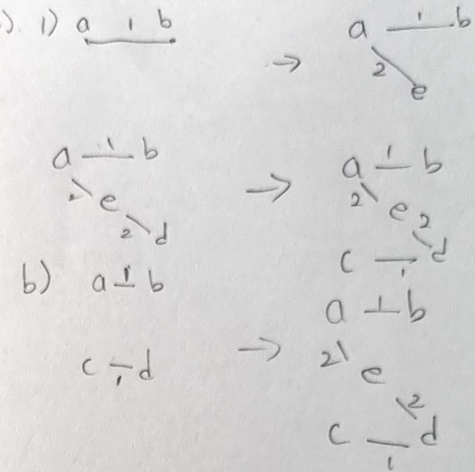
c)
```
Prod-Sum(int[] v, n)
  if(n==0) return 0;
  int[] OPT = new int [n+1];
  OPT[0]=0;
  OPT[1]=v[1];
  for int i=2 to n
    OPT[i]= max(OPT[i-1]+v[i], OPT[i-2]+v[i]×v[i-1]);
  return OPT[n];
```

2.
```
Least-Cents(int n)
  int coin[3]={1,10,25};   ← DP[0]=0
  for(int i=0; i<3; i++)
    for(int j=coin[i]; j<=n; j++)
      DP[j]=Min(DP[j], DP[j-coin[i]]+1)
  return DP[n]
```

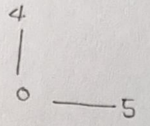
3. When it is 14 cents
In greedy algorithm {11,1,1,1} → 4coins
But the fewest coins is {7,7} → 2 coins
So, It does not always find fewest coin

4.
```
DP[0][0]=M[0][0]
for(int i=1; i<M; i++)
  DP[0][i] = DP[0][i-1]+M[0][i]
  DP[i][0] = DP[i-1][0]+M[i][0]
for(int a=1; a<M; a++)
  for(int b=1; b<N; b++)
    DP[b][a]=Max(DP[b-1][a]+M[b][a],
                 DP[b][a-1]+M[b][a])
return DP[N-1][M-1]
```

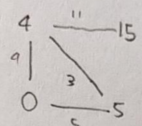
5. a). 1)



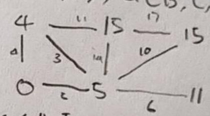
6.



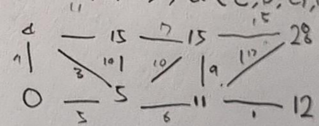
S<A>, Q(B,C,D,E,F,G,H)



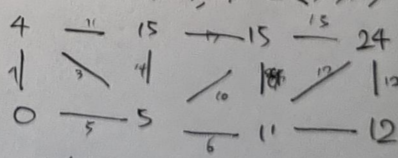
S<A,E>, Q(B,C,D,F,G,H>



S<A,E,B,F>, Q(C,D,G,H)



S<A,E,B,F,G,C> Q<D,H>



S<A,E,B,F,G,C,D,H> Q=∅

7. Gd ⊂ Gb is true
Gb is works with negative-weight edges and detects if there is a negative-weight cycle
but Gd is requires all edge weights to be nonnegative

So Gd ⊂ Gb