**Bitmask DP Approach for Set Cover Problem:**

This method will give you optimal minimum cost for weighted set cover problem. Here is the recurrence relation.  
  
N = number of subset

M = size of universe.

**if nowConsiderIndex == N & coveredMask!=(1<<M) - 1 :**  
I have considered all the elements before **nowConsiderIndex,** and I only covered the elements whose corresponding bit is on in coveredMask. And here I failed to cover all.  
so, f( coveredMask, nowConsiderIndex ) = Infinity.

**else if nowConsiderIndex == N & coveredMask==(1<<M) - 1 :**  
I have considered all the elements before **nowConsiderIndex,** And here I to covered all elements as all the N bits are on.

so, f( coveredMask, nowConsiderIndex ) = 0.

**else**

We have two options, we may take subset with id nowConsiderIndex or we may ignore. We will choose the optimal way.

valChoose= f( coveredMask | mask[nowConsiderIndex], nowConsiderIndex+1 ) + weight[nowConsiderIndex];

valNotChoose= f( coveredMask, nowConsiderIndex+1 );

f( coveredMask, nowConsiderIndex ) = min( valChoose, valNotChoose );