Part 1) Provide a pseudo-code for the algorithm to find a MWMCM for an edge-weighted CBG. Do not include the input parsing part; assume that you are given a CBG, and you can iterate over its left and right nodes, as well as over its edges.
Solution:
Assume there are n edges and we are iterating over all possible combinations of the edges.
Assume the fullEdgeSet (u,v,w) is given.

```
maxcar=0;
maxweight=0;
int[][] MWMCM;
for(i=1;i<2^n;i++){                                                    -- O(2^n)

        iBinary=Convert i to binary number of length n; (The ones in this binary number
determine what edges are present in this combination )                  -- O(n)
        numEdges=sumArray(iBinary);                                    -- O(n)
        edgeSetindex=Go over fullEdgeSet and get the indices of edges that are actually present;
-- O(n)
        weight=isMatching(fullEdgeSet, edgeSetindex);                  -- O(n)
        if(weight==0)  //weight=0 implies not a matching
                continue;
        //If we are here, then we have a matching
        car=2*numEdges;
        if(car<marcar)
                continue;
        elseif(car==maxcar)
                if(weight<=maxweight)
                        continue;
                else {maxweight=weight;
                        MWMCM=getEdges(fullEdgeSet,edgeSetindex);}     -- O(n)
        else if(car>maxcar){
                maxcar=car;
                maxweight=weight;
                MWMCM=getEdges(fullEdgeSet,edgeSetindex);              -- O(n)
                }

}
//After running this loop we will have maxcar, maxweight and the corresponding MWMCM
```

Part 2) Complexity of the code:
is $O(2^n)*(O(n)+O(n)+O(n)+O(n)+O(n))$
$=O(n2^n)$

This is the best we can do for this problem because finding all possible MWMCMs requires brute force approach. Using G-S Algorithm we can find one MWMCM, but if we want all MWMCM for a given CBG, we need to search over all possible over combinations, which is $O(2^n)$. The operations inside the main for loop, like finding is a combination is a matching, finding its weight etc require $O(n)$. So, the complexity of the pseudo code provided above is $O(n2^n)$.