Daren Liu

1) a) It is tight because it has an efficiency of $O(n)$

b) It is tight because the algorithm checks all elements of the upper triangular part of the adjacency matrix until a 0 is found or all elements are checked. $O(n^2)$

c) ~~###~~ Tight. The algorithm takes constant time on each subset. $O(2^n)$

d) Not tight. The lower bound of the algorithm is $O(n \log n)$.

2) a) We can find all the subarrays in polynomial time from 1c, which has a time complexity of 2^n. We can then find the sum and return true if their weights are equal.