Part1

Algorithm finds ratio of each weight and time. And after finding ratio, algorithm sorts weight and time lists respectively in order like the array of ratios.

In function **question1** which is in part1.py file, has a nested for loop each has n iteration and every iteration takes constant time. So, running time is **O(n^2)**

Part 2

a) Suppose n = 4, M = 10, and the operating costs are for NY [10,5,10,5] and for SF [5,10,5,10]

Then solution should be SF,SF,SF,SF and total cost 5+10+5+10 = 30. However, the given algorithm will give us SF,NY,SF,NY with costs 5+5+5+5+30=50.
So, the given algorithm does not correctly solve this problem by giving an instance which it does not return the correct answer.

b) optNY denotes the minimum cost of a plan on month in NY and optSF denotes the minimum cost of a plan on month in SF. If optimal plan ends in NY then it will pay cost of NY and plus the cost of the optimal plan n-1 months ending in NY or plus the cost of the optimal plan n-1 months ending in SF. If optimal plan ends in SF then it will pay cost of SF and plus the cost of the optimal plan n-1 months ending in NY or plus the cost of the optimal plan n-1 months ending in SF.

In function **question2** which is in part2.py file, function has n iterations and every iteration takes constant time. Running time is **O(n)**