문제접근.

각 가방은 최대 M kg, 최대 두 개의 가방이 들어갈 수 있는 캐리어는 최대 M kg까지 수용할 수 있다. 따라서 캐리어의 무게가 M kg를 넘어가지 않는 선에서, 가장 무거운 가방 하나와 가장 가벼운 가방 하나씩을 캐리어에 넣으며 캐리어를 구성해 나간다. 즉, 가장 무거운 가방의 무게와 가장 가벼운 가방의 무게의 합을 보고, 만약 M보다 크다면 가장 무거운 가방의 무게만 넣고, 그렇지 않다면 두 가방 모두 하나의 캐리어에 넣는다. 모든 가방이 캐리어에 들어갈 때까지 이 과정을 반복한다.

해결방법.

주어지는 가방들에 있어서, 무게가 같은 가방들은 모두 묶어 하나의 객체로 본다. 하나의 객체 bagType에는 가방의 무게와 총 개수가 입력된다. bagType들은 한 컨테이너(bags) 안에 구성하고, 무게 순으로 bagType은 정렬되어있다 하자. 또한, bags에서 현재 가장 가벼운 가방은 startBag, 가장 무거운 가방은 endBag이라 하자.

1. 입력을 받아 bagType을 생성하고, bags를 구성한다. 또한, 캐리어의 개수 carriers의 값은 0으로 초기화한다.
2. bags의 첫 번째 bagType을 startBag, 마지막 bagType을 endBag으로 초기화한다.
    a. M에서 endBag의 무게를 뺀 값이 startBag보다 크다면, 가장 무거운 가방과 함께 캐리어에 들어갈만한 가방이 없다는 것이므로, carriers의 값에 endBag의 개수를 더해준다. 이어서 endBag은 바로 앞 bagType으로 설정해준다.
    b. 만약 a에 해당하지 않는다면 startBag과 endBag의 개수 중 더 작은 값을 carriers에 더해준다. 그리고 그 값을 각 Bag의 개수에서 빼준다. 더 작은 개수를 가진 Bag은 모두 캐리어에 들어간 것이므로, 해당 Bag이 startBag이라면 그 다음, endBag이라면 그 직전 Bag으로 설정해준다.
        i. 만약 개수가 같은 경우라면 startBag과 endBag 각각을 그 다음, 그 직전 Bag으로 설정해준다.
    c. startBag과 endBagdl 동일해질 때까지 a와 b의 과정을 반복한다.
3. 최종적으로 구해진 carriers값이 곧 정답이다.

시간복잡도.

bagType들로 구성된 bags를 startBag과 endBag부터 하여 중간에 만날 때까지 조회하므로,
주어진 가방들에 있어서 최대로 구성될 수 있는 bagType의 수는 M개에 대해 O(M)