Competitive Programming SS24

Submit until end of contest

Problem: knapsack (1.0 second timelimit)

The Knapsack problem is a very well-known optimization problem: Given n items with size w_i and value p_i and a knapsack of size M, select some items which fit in the knapsack and maximize the total value of the selected items. In other words, find some $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} p_i$ is maximized under the additional constraint $\sum_{i \in S} w_i \leq M$.

In 1972, Richard Karp included this problem in his seminal paper "21 problems you won't believe are actually probably hard" and it made another appearance in the 2000 article "Solve one of THESE easy problems and become a millionaire!" by the Clay Mathematics Institute. As of April 29, 2021, no polynomial time solution for the Knapsack problem is known.

Year after year, students around the world rejoice when they discover an algorithm for the Knapsack problem that runs in time $\mathcal{O}(n \cdot M)$. However, when they demonstrate their algorithm to their professors using an example instance containing their bank account number, the professors ask them to leave their office, muttering something about "binary representation" and "pseudopolynomial algorithm", whatever that means.

Unfortunately, you also won't become a millionaire by solving this problem. But maybe you can still feel the joy seeing your solution actually give the correct answer. However, working on such an important problem is probably pretty intimidating for you, so we've decided to make the problem a little bit easier for you by setting $p_i = w_i$ for all i.

Input The input consists of a single testcase, which consists of two lines. The first line contains two integers, n and M ($1 \leq n, M \leq 2 \cdot 10^5$): The number of items and the capacity of the knapsack.

The second line contains n integers w_i ($1 \leq w_i \leq 3$) separated by single spaces: The weight of the items, which is also the value of the items.

Output Output a single integer: The value of an optimal solution to the Knapsack problem.

Sample input

```
3 5
3 3 1
```

Sample output

```
4
```