Robbing Gringotts

The Death Eaters are low on funds, and their leader Voldemort has asked them to get more money quickly, or face his wrath. They decide that the best way is to rob Gringotts Wizarding Bank.

By threatening one of the goblins in charge of the bank, the 'M' Death Eaters have discovered that Gringotts has 'N' vaults, with vault number 'i' containing X[i] gold items. The weights of the gold items in the ith vault are g[i][1],g[i][2],...,g[i][X[i]]. But as soon as a vault is robbed, the magical wards go off and alarm bells ring. Thus they have enough time to rob just one vault each, and all robberies have to take place at the same time. The Death Eaters have decided that no two of them will rob the same vault as this increases the chances of them being caught.

Death Eater j has a bag which can hold weight v[j]. They are a greedy and cowardly lot, so a Death Eater will only agree to rob a vault if he can fill up his bag completely to its capacity by taking some subset of the objects present in that vault. Note that it is not possible for a Death Eater to break any gold item; either it should be taken fully, or not be taken.

Find the maximum weight of gold they can take away by planning their strategy correctly.

Input (STDIN):

The first line contains the number of test cases T. T test cases follow. Each test case contains integers M and N on the first line. The next line contains M integers, the jth of which is v[j], which is the maximum weight of gold the jth Death Eater's bag can hold. The following N lines describe the gold items in the vaults. The ith line contains X[i], the number of gold items present in the ith vault, followed by X[i] integers g[i][1]..g[i][X[i]], denoting the weights of the each of the items present in the ith vault. There is a blank line after each test case.

Output (STDOUT):

Output one line for each test case, containing the maximum total weight of gold that the Death Eaters can rob.

Constraints:

1 <= T <= 10

1 <= N,M <= 50

1 <= X[i] <= 25

1 <= v[i],g[i][j] <= 10,000,000

Sample Input:

2

2 3

3 9

2 4 5

2 3 9

1 10

4 4

3 7 8 10

3 1 2 4

2 3 7

4 2 2 4 4

1 3

Sample Output:

12

28