拼图

【故事背景】

JYY 最近迷上了拼图游戏。作为一个计算机科学家，JYY 有一套黑白色的拼图，他希望通过合理的拼接，使得拼出的最终图案中，能包含面积最大的全白色子矩形。

【问题描述】

JYY 一共有 S 块拼图，并且由 1 到 S 编号。编号为 i 的拼图是一个 N 行 W_i 列的方格矩形，每个方格都为黑色或者白色。

一开始 JYY 将他的这 S 块拼图按照编号顺序左右相连依次放在桌上拼成了一个 N 行 M 列（这里 $M = \sum_{i=1}^S W_i$ ）的大矩形。

之后 JYY 发现，可以通过改变这 S 块拼图的连接次序，使得拼成的 N 行 M 列的大矩形中，最大全白色子矩形面积变大。

现在 JYY 想知道，怎么拼才能得到最大的全白色子矩形呢？请你帮助他计算出最佳的拼接方案。

【输入格式】

从文件 `puzzle.in` 中读入数据。

每个输入文件中包含多组测试数据。输入文件第一行包含一个整数 T ，代表测试数据的组数，接下来按顺序描述了每组测试数据。

每组测试数据的第一行包含两个整数 S 和 N 。

接下来 S 组输入，第 i 组对应编号为 i 的拼图。

在第 i 组输入中，第一行包含一个整数 W_i ；

接下来 N 行描述一个 N 行 W_i 列的 0/1 矩形；

其中第 x 行 y 列为 0 则表示该拼图对应位置的颜色是白色，反之则为黑色。

【输出格式】

输出到文件 `puzzle.out` 中。

对于每组数据输出一行包含一个整数 `ans`，表示最大可能的全白色子矩形的面积。

【样例输入】

```
1
3 4
4
1001
0000
```

```
0010
1001
3
000
010
000
011
2
00
10
01
00
```

【样例输出】

6

【样例说明】

最佳方案所最终拼成的大矩形为

```
000100100
010000010
000001001
011100100
```

粗体部分是面积为 6 的白色子矩形。

【数据规模与约定】

对于 10% 的数据满足 $S = 1, N \leq 30, W_i \leq 30$;

对于 30% 的数据满足 $S \leq 3$;

对于 50% 的数据满足 $S \leq 1000$;

对于 100% 的数据满足 $1 \leq S, N, W_i \leq 10^5, N \times \sum_{i=1}^S W_i \leq 10^5, T \leq 3$ 。