Getting started with Competitive Programming

Week 7 Practice Programming Problem

Problem: Long Distance Games

Similar Practice Problem - **https://www.hackerearth.com/practice/algorithms/graphs/depth-first-search/practice-problems/algorithm/maximum-spanning-tree/**

When your friends find out that you are doing a NPTEL course (that too on competitive programming!), they challenge you to a game, with the following rules.

- In **D**-dimensional Euclidean spaces, **N** lattice points are given. Every pair of points are not connected initially. The points are **not** guaranteed to be distinct.
- Let $(A_1, A_2, ..., A_D)$ and $(B_1, B_2, ..., B_D)$ be the coordinates of the points **A** and **B** respectively, then the distance between **A** and **B** is defined as $((A_1-B_1)^2 + (A_2-B_2)^2 + ... + (A_D-B_D)^2)^{1/2}$.
- Your initial score is **one**.
- You can play as much as you want (of course, terms and conditions apply; if you have no valid moves left then you cannot take any more turns).
- In every turn, you can connect an unconnected pair of two given points, if this new connection does not form a cycle. That is, you cannot connect the pair of points **A** and **B**, if there exist points X_1, X_2, ..., X_k such that **A** connected to X_1, X_1 connected to X_2, X_2 connected to X_3,..., and X_k connected to **B**.
- Every time that you connect a pair of points, your score is multiplied by the square of the distance between them.

Fortunately, this challenge happens during the very week that you are studying spanning trees, and of course, you are going to win! Specifically, you are going to determine the largest score that you can obtain. Since this number can be huge, you should output it modulo **747474747**.

Input

The first line of input contains **T**, denoting the number of test cases. Then **T** test cases follow.

The first line of each test case contains two space-separated-integers **N** and **D**. The next **N** lines contain **D** space-separated integers, denoting the coordinates of the given lattice points.

Output

For each test case, output the maximal score modulo **747474747**.

Constrains

- $1 \leq T \leq 6666$
- $1 \leq N \leq 6666$
- $1 \leq D \leq 5$
- The absolute value of any integer given in the input does not exceed **100000000** (10^8).
- The sum of **N** in one input file does not exceed **6666**.

Example

Input:

```
1
3 2
0 0
-1 -1
1 -1
```

Output:

```
8
```

Explanation

The distance between the first point and the second point is $\mathbf{((0-(-1))^2+(0-(-1))^2)^{1/2} = 2^{1/2}}$. The distance between the first point and the third point is $\mathbf{((0-1)^2+(0-(-1))^2)^{1/2} = 2^{1/2}}$. The distance between the second point and the third point is $\mathbf{(((-1)-1)^2+((-1)-(-1))^2)^{1/2} = 2}$.

One of the optimal ways is that you connect the third and the second points, and then connect the first and the second points. The maximum score is $\mathbf{2^2 * (2^{1/2})^2 = 8}$.

Problem 2 - https://www.codechef.com/ICL2016/problems/ICL16A