# Applications of Randomization in OI

**Translation by Kirito Feng**

LIU JIAYU

CTSC 2007

## Contents

## Abstract

Nowadays, the topics of information science competition are changing with each passing day, and various new algorithms emerge in an endless stream. As an emerging algorithm, randomization algorithms play a unique and important role in the field of informatics competition. This paper briefly introduces the application of randomization algorithm in informatics competition through several typical examples.

## §1 Introduction

Along with the development of the informatics competition, the term "randomization" experienced a process from strange to familiar in the hearts of students participating in the informatics competition. With its unique flexibility and variety, the randomization algorithm is gradually used in more and more different types of competition topics, and has unique advantages in the increasingly diverse informatics competition. Mastering the randomization algorithm, there is no doubt that there is a more problem-solving blade.

## §2 Sample Problems

### §2.1 Geometrical dreams (Ural1046)

There is a polygon $A_1A_2 \ldots A_N$, and an isosceles triangle $A_iM_iA_{i+1}$ is constructed outside the polygon on each side $A_iA_{i+1}$ such that the angle $A_iM_iA_{i+1} = \alpha_i$. The set consisting of $\alpha_i$ satisfies the angle of any non-empty subset thereof and is not a multiple of 360 degrees. Given $N$, the coordinates of all $M_i$ and $\alpha_i$, write a program that outputs the coordinates of the vertices of the polygon.

#### §2.1.1 Analysis

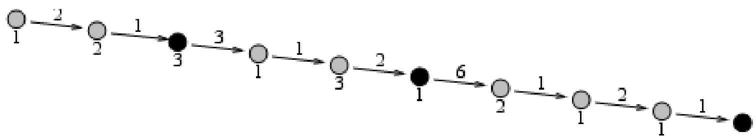
This problem can be solved with a simple solution equation. Let us think about the problem from another angle. From the condition of the title, as long as the coordinates of a vertex are determined, the coordinates of other vertices of the polygon can be obtained by simple calculation, then the problem is transformed into the coordinates of a vertex of the determined polygon. How to determine the coordinates of a vertex, common algorithms such as enumeration and binary can not solve the problem for us, so we think of randomization. We start by placing the first point at the origin, and we can get the coordinates of the $N+1$th vertex by calculation. If the $N+1$th vertex and the first vertex coincide, this polygon is what we want, but obviously The possibility is very embarrassing, so we need to adjust the position of the first point. Obviously, the smaller the distance of the first point from the $N+1$th point, the closer its position is to its actual position. Each time we can randomly determine a point near the position of the first point, and determine whether the distance between the first point and the $N+1$th point is greater than the original when the first point is placed at this position. Small, if it is, then the position of the first point is temporarily set at this position, and then continue the above operation until the first point coincides with the $N+1$th point, and we determine the coordinates of the vertices of the polygon.

It turns out that this method can pass this topic. Although this method is obviously more complicated in any respect than the common practice mentioned above, it does not make much sense to solve this problem, but it provides us with a new idea - randomization. The randomization algorithm has no advantage for such a problem, but it can be used on many problems. Let's take a closer look at the charm of the randomization algorithm.

## §2.2 Two Sawmills (CEOI2004)

There are $n$ old trees on the road from the top of the mountain to the foot of the mountain. Now the government has decided to cut them down. In order not to waste wood, every tree will be transferred to the sawmill.

The tree can only be transported in one direction, down. There is a sawmill at the end of the road. Two additional sawmills can be located on any of the old trees on the road, and you must choose where to build them to minimize shipping costs. Shipping costs are one cent per meter per kilogram of wood.



### §2.2.1 Analysis

The standard algorithm for this topic converts the data into an image, which is processed by the stack to find the maximum coverage area of the two rectangles, with a time complexity of $\mathcal{O}(N)$. However, this algorithm is not too small to be able to think about.

Let's look at the performance of the randomization algorithm on this topic.

First of all, the most easily thought of randomization is of course to directly search for two points, calculate the total freight when the two points are sawmills, and output the minimum cost after multiple randomizations.

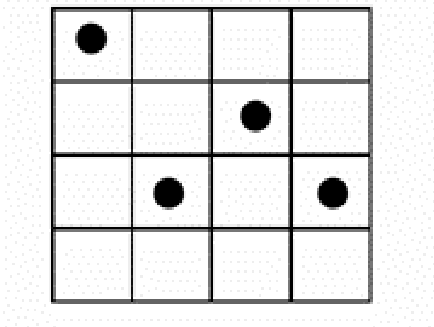
We can pre-process and reduce the computational time complexity to $\mathcal{O}(1)$, so we can randomly or several tens of millions of times within the time limit, but find the most relative to the total state of 400 million. The chances of a good solution are not great.

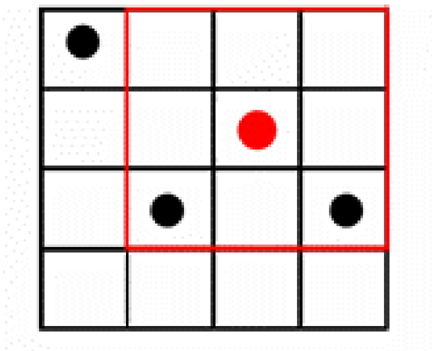
Is there a better way?

We just used the randomization algorithm to solve the problem directly. The accuracy is not very good. In order to increase the accuracy, we try to use randomization to narrow the area.

We create a matrix $P$, $P[X, Y]$ represents the total freight cost of the first sawmill established at $X$ and the second sawmill established at $Y$. In the beginning, the side length of the matrix is $N$. We randomly look for a certain number of points (as shown in the left figure below, the number of points should make full use of the time limit and pay attention to efficiency, since the size of the matrix has been changing, it is recommended to use the ratio of the matrix size to determine the number of points), calculate their Value, take its minimum point, take this point as the center of the new matrix, take the length of $\frac{3}{4}$ of the side length of the current matrix as the side length of the new rectangle (as shown in the right figure below), take a piece from the original matrix as The range of the new matrix (if the range of the new matrix is out of the boundary of the original matrix, move it inside to the original matrix), and then continue to repeat this operation in the new matrix until the new matrix is small enough, we can enumerate For each point on the new matrix, take the minimum value as the answer.

We were pleasantly surprised to find that this randomization algorithm can pass all the test data!





It can be seen from this topic that the flexibility of the randomization algorithm makes it have a wider range of applications, and in many places that seem difficult to start by subtly using the radiant heat. And this variability also makes us need to use the randomization algorithm flexibly and properly to take advantage of it. The randomization algorithm is not just a random mess. When using a randomization algorithm, it is worth considering as much as other algorithms. Let's take a look at the application of the randomization algorithm in the actual game.

## §2.3 Example - Small H's Party (NOI2005)

Little H liked computers very much since he was a child. After he was in middle school, he was even fascinated with computer programming. After years of unremitting efforts, Little H was lucky to be selected into the information competition provincial team. He was going to participate in the 22nd National Informatics Olympiad (NOI2005) in Zhengzhou, Henan Province.

Little H's good friends Little Y and Little Z got the news and they are sincerely happy for him. They are going to host a party, invite Little H and all his friends to participate and celebrate for Little H. After several days of investigation, Little Y and Little Z listed a list of all the friends of Little H. There are a total of N people (for convenience, we will number them as 1 to $N$). However, there are too many people on the list, and many of them are not known to Little Y and Little Z. How to organize them all to join the party?

Little Y and Little Z hope to design a network of contacts for $N$ friends of Little H, so that if someone knows about the latest situation of the party, others can get the message directly or indirectly. At the same time, in order to ensure the message delivery is simple, efficient and the most important point: confidentiality (in order to give a small H a surprise, the news of the party at the party's preparatory stage is absolutely impossible to let him know), small Y and small Z decided Let the least number of friends contact directly: In order to ensure that $N$ friends can directly or indirectly contact each other, just let $(N-1)$ contact the friend directly.

Obviously, the friends on the list don't all know each other, and even the two people who know each other have different levels of familiarity. Therefore, according to the results of the survey, Little Y and Little Z listed a relationship table between friends. The table indicates which people can be directly contacted, and for each pair of friends who can contact each other, Little Y and Little Z They also marked the level of enjoyment of the contact. For example, the relationship between 3 and 4 is very good, so the relationship between them is 10; and 1 and 3 are average friends, then they are less pleasant. Figure 1 above shows a contact list of N=5, where the points represent the friends on the list, and the side indicates that the two friends can directly contact, and the numbers on the side are the pleasure of their contact.

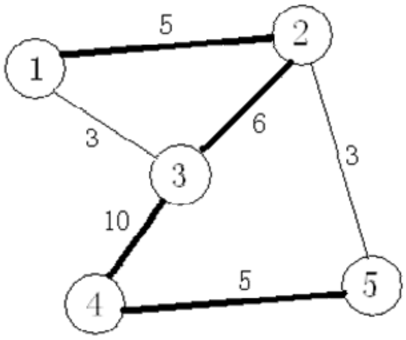



图 1

Little Y and Little Z hope that everyone can enjoy this party, so they decided to maximize the enjoyment of the network: the so-called happiness of the network, the sum of the happiness of each pair of direct contacts. As in Figure 1, the bolded edges represent a network that maximizes the level of enjoyment, with a pleasantness of $5+6+10+5=26$. However, if you let someone directly contact a lot of people, this will inevitably put a big burden on him. Therefore, Little Y and Little Z also set a maximum number of direct contacts for each person, indicating that in the contact network, at most $k_i$ individuals can directly contact $i$. Still using the example of Figure 1, if we add $k_i = 1, 1, 4, 2, 2$ to each of 1 to 5, then the above solution will not meet the requirements. The optimal scheme at this time is shown in Fig. 2, and the degree of enjoyment is $3+6+10+5=24$.

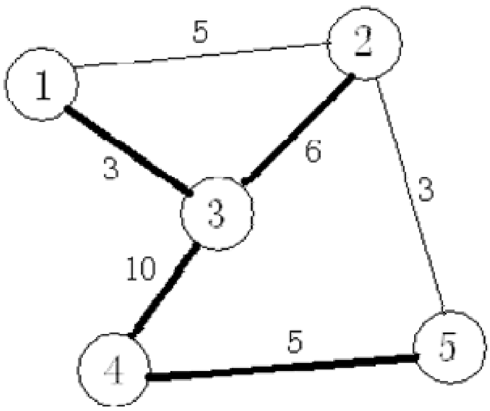



图 2

Can you help Little Y and Little Z find a contact network that is as enjoyable as possible under the premise of satisfying the restrictions?

### §2.3.1 Analysis

This is an answer to the question, the test data is divided into three types, the first type includes the first 1-3 data, the degree limit of each point is $N-1$, equivalent to no limit, directly using the minimum spanning tree algorithm Can be solved. The second type includes the 4th to 6th datasets, where the degree of $N-1$ points is limited to $N-1$, and only one point has a degree limit, which can be solved using the minimum limit spanning tree algorithm.

The third type of topic data includes the 7-10th data. All points in the data have degree limits, which are NP problems, and this topic is an open topic that does not require an optimal solution. This kind of problem is really a vast sky in which the randomized algorithm is free to fly.

We can use a randomization algorithm combined with different algorithms to solve this problem.

**Method 1: Kruskal-based randomization algorithm**

We use the Kruskal algorithm to calculate a spanning tree that meets the degree limit of the point. Because of the degree limitation, the resulting spanning tree is not necessarily optimal. We can sort it by the marginal weight and then randomly order the chaotic parts is followed by the Kruskal algorithm in order to obtain a better solution.

After each such large spanning tree is obtained, we randomly select an edge that is not on the spanning tree, add a cycle to the spanning tree, and then try to delete an edge on the cycle so that the spanning tree is still satisfied. Limitation and better results, so repeated multiple times, to get a superior solution that can no longer be better solved by this method.

The Kruskal algorithm is used multiple times to output the calculated optimal solution.

**Method 2: Prim-based randomization algorithm**

First, use Prim algorithm to calculate a maximum spanning tree with no consideration limit. On this spanning tree, each time a point with a non-satisfaction limit is randomly selected, and an attempt is made to find a point that does not replace a side in the spanning tree and is connected to the point in the spanning tree to lower the point. It stops until the tree is modified to the satisfaction limit or the degree of the point where the tree is not satisfied is reduced by such a method.

Repeat the above operation multiple times on the original maximum spanning tree to output the calculated optimal solution.

It can be seen from the above examples that the use of randomization algorithms can achieve good results on open topics, and the use of randomization combined with different algorithms will yield different results. Appropriate algorithms and randomization algorithms should be used in solving problems to improve the accuracy of the program.

## §3 Conclusion

As can be seen from the above questions, the randomization algorithm has the characteristics of flexibility and change. The application of the randomization algorithm should pay attention to the following points:

- Use randomized algorithms flexibly for different situations to give full play to its flexible and changeable features
- Pay attention to the application of randomization algorithm combined with appropriate algorithm to stimulate greater vitality
- Pay attention to the efficiency of random decision and adjustment, and make full use of the time limit given by the problem

As an emerging algorithm with distinctive features, the randomization algorithm not only provides us with different methods to solve many problems, but also has obvious advantages in open topics. The flexible mastery and use of randomized algorithms will undoubtedly make us more comfortable in the changing information science field.