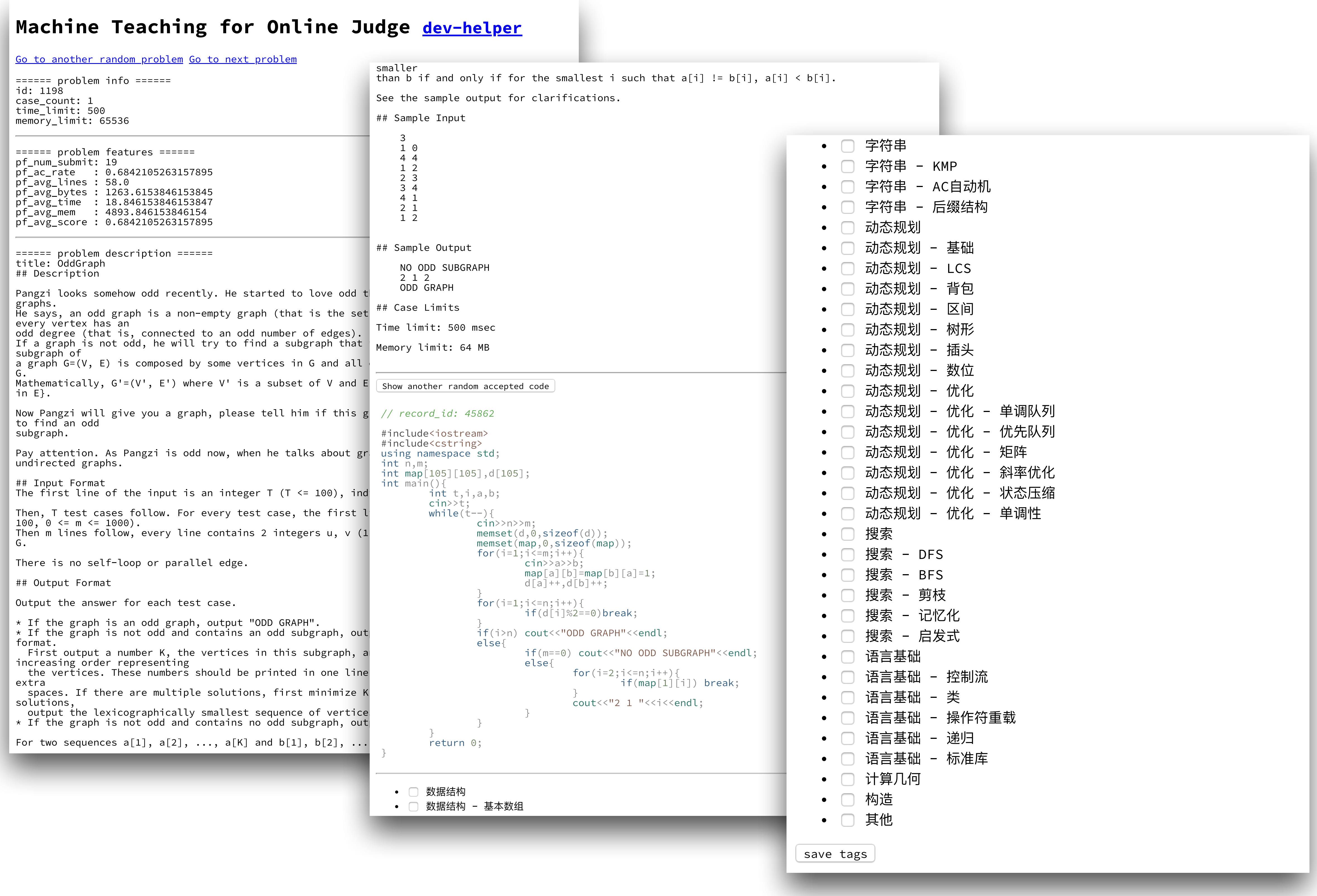

Machine Teaching for Online Judge dev-helper

Go to another random problem Go to next problem

```
====== problem info ======
id: 1198
case_count: 1
time_limit: 500
memory_limit: 65536
```

```
====== problem features ======
pf_num_submit: 19
pf_ac_rate   : 0.6842105263157895
pf_avg_lines : 58.0
pf_avg_bytes : 1263.6153846153845
pf_avg_time  : 18.846153846153847
pf_avg_mem   : 4893.846153846154
pf_avg_score : 0.6842105263157895
```

```
====== problem description ======
title: OddGraph
## Description

Pangzi looks somehow odd recently. He started to love odd t
graphs.
He says, an odd graph is a non-empty graph (that is the set
every vertex has an
odd degree (that is, connected to an odd number of edges).
If a graph is not odd, he will try to find a subgraph that
subgraph of
a graph G=(V, E) is composed by some vertices in G and all 
G.
Mathematically, G'=(V', E') where V' is a subset of V and E
in E}.

Now Pangzi will give you a graph, please tell him if this g
to find an odd
subgraph.

Pay attention. As Pangzi is odd now, when he talks about gr
undirected graphs.

## Input Format
The first line of the input is an integer T (T <= 100), ind

Then, T test cases follow. For every test case, the first l
100, 0 <= m <= 1000).
Then m lines follow, every line contains 2 integers u, v (1
G.

There is no self-loop or parallel edge.

## Output Format

Output the answer for each test case.

* If the graph is an odd graph, output "ODD GRAPH".
* If the graph is not odd and contains an odd subgraph, out
format.
  First output a number K, the vertices in this subgraph, a
increasing order representing
  the vertices. These numbers should be printed in one line
extra
  spaces. If there are multiple solutions, first minimize K
solutions,
  output the lexicographically smallest sequence of vertice
* If the graph is not odd and contains no odd subgraph, out

For two sequences a[1], a[2], ..., a[K] and b[1], b[2], ...
```

```
smaller
than b if and only if for the smallest i such that a[i] != b[i], a[i] < b[i].

See the sample output for clarifications.

## Sample Input

    3
    1 0
    4 4
    1 2
    2 3
    3 4
    4 1
    2 1
    1 2


## Sample Output

    NO ODD SUBGRAPH
    2 1 2
    ODD GRAPH

## Case Limits

Time limit: 500 msec

Memory limit: 64 MB
```

Show another random accepted code

```
// record_id: 45862

#include<iostream>
#include<cstring>
using namespace std;
int n,m;
int map[105][105],d[105];
int main(){
	int t,i,a,b;
	cin>>t;
	while(t--){
		cin>>n>>m;
		memset(d,0,sizeof(d));
		memset(map,0,sizeof(map));
		for(i=1;i<=m;i++){
			cin>>a>>b;
			map[a][b]=map[b][a]=1;
			d[a]++,d[b]++;
		}
		for(i=1;i<=n;i++){
			if(d[i]%2==0)break;
		}
		if(i>n) cout<<"ODD GRAPH"<<endl;
		else{
			if(m==0) cout<<"NO ODD SUBGRAPH"<<endl;
			else{
				for(i=2;i<=n;i++){
					if(map[1][i]) break;
				}
				cout<<"2 1 "<<i<<endl;
			}
		}
	}
	return 0;
}
```

- □ 数据结构
- □ 数据结构 - 基本数组

- □ 字符串
- □ 字符串 - KMP
- □ 字符串 - AC自动机
- □ 字符串 - 后缀结构
- □ 动态规划
- □ 动态规划 - 基础
- □ 动态规划 - LCS
- □ 动态规划 - 背包
- □ 动态规划 - 区间
- □ 动态规划 - 树形
- □ 动态规划 - 插头
- □ 动态规划 - 数位
- □ 动态规划 - 优化
- □ 动态规划 - 优化 - 单调队列
- □ 动态规划 - 优化 - 优先队列
- □ 动态规划 - 优化 - 矩阵
- □ 动态规划 - 优化 - 斜率优化
- □ 动态规划 - 优化 - 状态压缩
- □ 动态规划 - 优化 - 单调性
- □ 搜索
- □ 搜索 - DFS
- □ 搜索 - BFS
- □ 搜索 - 剪枝
- □ 搜索 - 记忆化
- □ 搜索 - 启发式
- □ 语言基础
- □ 语言基础 - 控制流
- □ 语言基础 - 类
- □ 语言基础 - 操作符重载
- □ 语言基础 - 递归
- □ 语言基础 - 标准库
- □ 计算几何
- □ 构造
- □ 其他

save tags