修修改

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

— 我们要修、修改 a、a_i…… —

给定一个长度为 n 的正整数序列 $a_1, a_2, \ldots, a_n$，满足该序列单调不减，即 $a_1 \le a_2 \le \cdots \le a_n$。
你可以进行若干次如下操作：

- 选择一个下标 i，满足 $1 \le i \le n$ 且 $1 \le a_i \le n$；
- 将 a_{a_i} 增加 1，即执行：

$$a_{a_i} \leftarrow a_{a_i} + 1.$$

请判断是否存在一种操作方案，使得最终序列 a 成为一个长度为 n 的排列。如果存在，请输出任意一种合法的操作方案。
一个长度为 n 的排列是指由 $1, 2, \ldots, n$ 这 n 个整数各出现一次组成的序列。

Input

第一行包含一个整数 T $(1 \le T \le 1000)$，表示测试数据组数。对于每组测试数据：

- 第一行包含一个整数 n $(1 \le n \le 1000)$，表示序列长度。
- 第二行包含 n 个正整数 $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 2000)$，表示初始序列。保证该序列单调不减，即满足 $a_1 \le a_2 \le \cdots \le a_n$。

对于所有测试数据，保证 $\sum n \le 1000$。

Output

对于每组测试数据：

- 如果存在合法方案，输出一行一个字符串 `YES`，随后输出两行，表示你给出的操作方案，其中：
  - 第一行输出一个整数 K，表示操作次数。要求 $0 \le K \le \frac{(n-1)n}{2}$。
  - 第二行输出 K 个整数 $p_1, p_2, \ldots, p_K$，表示依次选择的下标。每个 p_j 必须满足 $1 \le p_j \le n$，并且在执行第 j 次操作前必须有 $1 \le a_{p_j} \le n$。
  - 要求执行完这 K 次操作后，序列 a 必须是一个长度为 n 的排列。如果有多种合法方案，输出任意一种即可。
- 否则，如果不存在合法方案，只需输出一行 `NO`。

Example

standard input	standard output
3 3 1 1 1 4 1 1 3 3 4 2000 2000 2000 2000	YES 3 1 1 1 YES 2 3 1 NO

Note

- 对于第一组数据，序列变化情况为 $[1,1,1] \to [2,1,1] \to [2,2,1] \to [2,3,1]$
- 对于第二组数据，序列变化情况为 $[1,1,3,3] \to [1,1,4,3] \to [2,1,4,3]$