Shopee Code League 2022 - Qualification Round

Mar 19, 2022, 03:00 PM SGT - Mar 19, 2022, 06:15 PM SGT

Connecting the Numbers

Max. score: 100

This problem is no longer available for practice. Apology for any inconvenience!

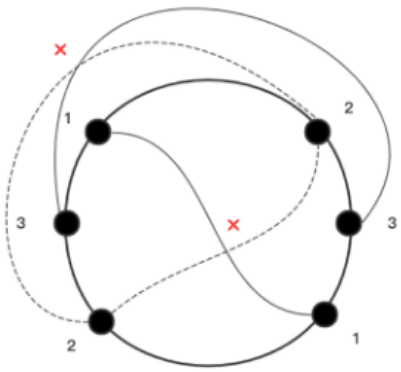
There are *2 * n* points on the circle which are on the two-dimensional plane. Each point has a *1* to *N* number and each number appears twice. The same numbers should be connected, but the following restrictions should be met:

1) Lines cannot intersect

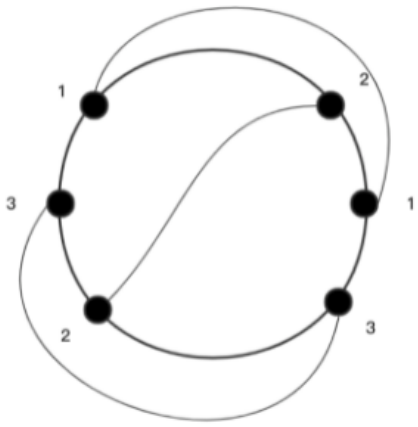
2) Select to connect outside or inside the circle

Sample

case 1:



case 2:



?

Input Format

The first line contains one integer *t* ($1 \leq t \leq 50$) - the number of test cases. Each test case consists of two lines:

The first line contains one integer *n* ($2 \leq n \leq 10^5$) - the number of points.

The second line contains *2*n* integers *ni* ($1 \leq i \leq 2 * n, 1 \leq n_i \leq n$) - the number of clockwise points.

Output Format

For each test case, print yes if there is a solution. Otherwise, print no.

SAMPLE INPUT

```
2
3
1 2 3 1 2 3
3
1 2 1 3 2 3
```

SAMPLE OUTPUT

```
no
yes
```

Explanation

NA

Time Limit:	2.5 sec(s) for each input file.
Memory Limit:	256 MB
Source Limit:	1024 KB
Marking Scheme:	Score is assigned when all the testcases pass.
Allowed Languages:	Bash, C, C++, C++14, C++17, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, Java 14, JavaScript(Rhino), JavaScript(Node.js), Julia, Kotlin, Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, Python 3.8, Racket, Ruby, Rust, Scala, Swift-4.1, Swift, TypeScript, Visual Basic

CODE EDITOR

Save Python 3 (python 3.9.5)

```
#!/usr/bin/env python3

import numpy as np

N = int(input())

def whack(A, B):
    a0, a1, b0, b1 = A[0], A[1], B[0], B[1]

    def between(a, b, c):
        return a < b < c
    def four(a, b, c, d):
        return a < b < c < d

    if between(a0, b0, a1) and between(a0, b1, a1): return False
    if between(a0, a1, b0) and between(a0, a1, b1): return False
    if between(b0, a0, a1) and between(b1, a0, a1): return False

    if four(b0, a0, a1, b1): return False
    if four(b1, a0, a1, b0): return False
```