

构造三角形

题意

给定 $n(n \geq 3)$ 根木棍，木棍的长度范围是 $[l, r]$ ，问：根据这些木棍，是否一定能选出三根构造出一个三角形？

题解

思考一下， n 根木棍在什么情况下无法构造出三角形？

若 n 根木棍从小到大的长度依次为： a_1, a_2, \dots, a_n ，即 $\forall i \in [1, n-1], a_i \leq a_{i+1}$ 。任选三根木棍的长度为 a_i, a_j, a_k ($1 \leq i < j < k \leq n$)。当此三根木棍无法构造出三角形时，必定存在 $a_i + a_j \leq a_k$ 。因而当 n 根木棍无法构造出一个三角形时，必定存在如下关系： $a_{i-2} + a_{i-1} \leq a_i$ ($i \geq 3$)。

在最坏情况下，当给定木棍无法构造出一个三角形时，它们的数学关系满足如下条件：

$$a_i = \begin{cases} l, & i \leq 2 \\ a_{i-2} + a_{i-1}, & i \geq 3 \end{cases} \quad (1)$$

当 $a_n \leq r$ 时，我们根据这些木棍无法构造出一个三角形，否则我们一定能够构造出一个三角形。

显然， a_i 具备指数级别的增长速度。和斐波那契数列相似， $n \geq 80$ 时， a_n 已超过数据范围。

判定最坏情况下无法构造出三角形的时间复杂度为： $O(T * \log(r))$ ，显然符合时间限制。

注： a_i 的增长速度为指数级别，当 $a_i > r$ 时应直接`break`，否则可能会爆`long long`。

Code

C\C++

```
1  #include<stdio.h>
2  typedef long long i64;
3
4  int calc(i64 l,i64 r){
5      int res = 3;
6      i64 a = 1,b = 1,c = 1 + 1;
7      while(c<=r){
8          i64 t = b + c;
9          a = b,b = c,c = t;
10         res++;
11     }
12     return res;
13 }
14
15 int main(){
16     int T; scanf("%d",&T);
17     while(T--){
18         i64 k,l,r;
19         scanf("%lld %lld %lld",&k,&l,&r);
20         printf("%s\n",k>=calc(l,r)?"YES":"NO");
21     }
22 }
```

Python

```
1  import sys
2
3  input = lambda: sys.stdin.readline().strip()
4  def main():
5      import sys
```

```
6 data = sys.stdin.read().split()
7 T = int(data[0])
8 ptr = 1
9 res = []
10 for _ in range(T):
11     k = int(data[ptr])
12     l = int(data[ptr + 1])
13     r = int(data[ptr + 2])
14     ptr += 3
15     # 构建最大三角形无法成立的集合
16     f = 2 # 初始有两个元素 1, 1
17     s1 = 1
18     s2 = 1
19     while True:
20         s3 = s1 + s2
21         if s3 > r:
22             break
23         f += 1
24         s1 = s2
25         s2 = s3
26     if k > f:
27         res.append("YES")
28     else:
29         res.append("NO")
30     print('\n'.join(res))
31
32 if __name__ == "__main__":
33     main()
```