

Spiel

Herr T. Rainer möchte ein Spiel mit dir spielen! Er hat sich eine Zahl zwischen 1 und N überlegt (N ist hierbei *gerade*), die du erraten sollst. Dazu kannst du ihm bis zu Q Fragen stellen. Für jeder der Fragen darfst du dir eine Teilmenge der Zahlen von 1 bis N aussuchen und Herr Rainer sagt dir, ob seine Zahl darin enthalten ist.

Als geübter Olympionike weiß er, dass es so viel zu einfach ist auf seine Zahl zu kommen. Deshalb hat Herr Rainer beschlossen, dass er bei *genau einer* der Q Frage lügen wird und dir die falsche Antwort gibt. Kannst du seine Zahl trotzdem erraten?

Implementiere dazu die Funktion `guess(N, Q)`, die nach maximal Q Fragen Herrn Rainers Zahl zurückgeben soll. Du kannst mithilfe der Funktion `ask(S)` eine Frage simulieren, wobei S eine Teilmenge der Zahlen von 1 bis N ist. Diese Funktion braucht $O(|S|)$ Zeit, wobei $|S|$ die Anzahl an Elemente in S sind. Interagiere (wie üblich) *nicht* mit dem Standardoutput.

Subtasks

Allgemein gilt:

- $1 < N \leq 10^5$ und N ist gerade
- Q ist so gewählt, dass es eine Strategie gibt Herrn Rainers Zahl zu erraten, egal wie er antwortet. D.h. wenn man die Fragen optimal nutzt, dann kommt man mit Q Fragen aus.

Subtask 1 (1 Punkte): Er lügt bei der ersten Antwort

Subtask 2 (2 Punkte): $2^Q \geq 4N^2$

Subtask 3 (31 Punkte): N ist eine Zweierpotenz (2, 4, 8, ...) und Q so klein wie möglich

Subtask 4 (42 Punkte): Q ist so klein wie möglich

Subtask 5 (24 Punkte): Keine Einschränkungen

Beschränkungen

Zeitlimit: 1 s

Speicherlimit: 256 MB