

Tennisturnier

Jedes Jahr spielen die Junioren der Enfield Tennis Academy (ETA) ein Turnier gegen die Konkurrenz von der Port Washington Academy.

In diesem Jahr ist Port Washington als Titelverteidiger Gastgeber des Turniers – zum Ausgleich darf ETA-Cheftrainer Gerhardt Schtitt die einzelnen Begegnungen festlegen. Genauer stellt jede der beiden Tennisakademien eine Mannschaft aus N Spielern ($1 \leq N \leq 750\,000$) und es sind N Begegnungen zwischen je einem ETA- und einem Port Washington-Spieler auszutragen, sodass jeder Spieler genau ein Match bestreitet. Gesamtsieger des Turniers ist dabei natürlich die Mannschaft mit mehr Siegen. Da der Gewinn des Turniers mit hohem Prestige (und umgekehrt die Niederlage mit nie endender Schmach) verbunden ist, möchte Schtitt eine möglichst kluge Wahl der Begegnungen treffen. Glücklicherweise hat er dazu die aktuelle Juniorenrangliste zur Verfügung und er ist überzeugt, dass diese den Ausgang einer jeden Begegnung exakt vorhersagen kann: ein Spieler gewinnt genau dann gegen einen anderen Spieler, wenn er in der Rangliste besser (d.h. mit niedrigerer Rangnummer) platziert ist. Insbesondere würde also die aktuelle Nr. 1 gegen alle anderen Spieler gewinnen. Wir gehen ferner davon aus, dass keine zwei Spieler gleichauf platziert sind.

Hilf Schtitt, indem du ein Programm schreibst, das, gegeben die Spieler der jeweiligen Mannschaften sowie ihre Platzierungen, entscheidet, welche Begegnungen Schtitt ansetzen soll, damit die ETA-Mannschaft die maximale Anzahl an Siegen einfährt.

Eingabe

Die erste Zeile der Eingabe enthält die ganze Zahl N mit der Bedeutung wie oben. Die nächste Zeile enthält N durch Leerzeichen getrennte positive ganze Zahlen r_i ($1 \leq r_i \leq 10^9$): die Ranglistenplatzierungen der ETA-Spieler. Darauf folgt eine weitere Zeile im selben Format mit den Platzierungen der Port Washington-Spieler.

Ausgabe

Die erste Zeile deiner Ausgabe soll die maximale Anzahl an Begegnungen, die die ETA-Mannschaft im Turnier gewinnen kann, enthalten. Es sollen N Zeilen folgen, welche eine optimale Ansetzung der Begegnungen (gemäß obiger Regeln) beschreiben. Dabei soll jede Zeile aus zwei ganzen Zahlen a und b bestehen ($1 \leq a, b \leq n$): das bedeutet, dass der a -te Spieler der ETA gegen den b -ten Spieler von Port Washington spielen soll (jeweils 1-basiert, gemäß der Reihenfolge in der Eingabe).

Wenn es mehrere Lösungen gibt, darf dein Programm eine beliebige davon ausgeben.

Beschränkungen und Bewertung

Stets gilt $1 \leq N \leq 750\,000$ sowie $1 \leq r_i \leq 10^9$. Ferner ist kein Ranglistenplatz mehrfach vergeben.

Teilaufgabe 1 (20 Punkte). $N \leq 10$.

Teilaufgabe 2 (40 Punkte). $N \leq 10\,000$.

Teilaufgabe 3 (40 Punkte). Keine weiteren Beschränkungen.

Darüber hinaus gilt: Ist in einer Teilaufgabe immer die erste Zeile deiner Ausgabe korrekt (aber in mindestens einem der entsprechenden Testfälle nicht der Rest), so erhältst du dafür 75% der Punkte.

Beispiele

Eingabe	Ausgabe
3 6 3 4 2 8 5	2 1 1 2 2 3 3
3 4 5 6 1 2 3	0 1 1 2 2 3 3

Limits

Zeit: 0.7 s

Speicher: 128 MiB

Feedback

Für diese Aufgabe ist *restricted feedback* verfügbar. Das bedeutet, die angezeigte Punktzahl entspricht der endgültigen Punktzahl deiner Einsendung. Allerdings wird dir für jede Testfallgruppe immer nur der erste Testfall mit minimaler Punktzahl innerhalb der entsprechenden Gruppe angezeigt. (Hierbei ist die Reihenfolge der Fälle innerhalb der jeweiligen Gruppen fest.)

Bewertung

Deine endgültige Punktzahl für diese Aufgabe ist die Summe über jede Teilaufgabe Deiner besten Punktzahl auf diese Teilaufgabe unter allen Deinen Einsendungen.