

Gra *agar.io* (swego czasu dość popularna w sieci) jest rozgrywana w przeglądarce internetowej. Każdy z obecnych na serwerze graczy steruje pojedynczą komórką o pewnej masie. Kiedy spotyka komórkę o masie mniejszej niż swoją, wchłania ją, powiększając swoją masę o masę drugiej komórki. Kiedy jednak spotka komórkę o masie większej lub równej swojej, natychmiast przegrywa.

Bajtek połączył się właśnie z serwerem gry i jego komórka ma niezbyt imponującą masę 2. Bardzo chciałby jednak wygrać, czyli zostać największą komórką na tym serwerze. Dzięki niebывалым umiejętnościom hakerskim udało mu się zablokować możliwość poruszania się innych graczy – czyli jest w stanie swobodnie wybierać kolejne ofiary do wchłonięcia, a inni gracze nie mogą uniknąć spotkania.

Ta miła sytuacja nie potrwa jednak długo i administratorzy serwera niebawem cofną blokady Bajtka. Bajtek jest w stanie wchłonąć swoją komórką inną komórkę w czasie jednej sekundy. Ile sekund będzie potrzebował, aby stać się największą komórką na serwerze? Bajtek zadowolony ewentualnym remisem z innymi największymi komórkami.

Napisz program, który na podstawie masy komórek kontrolowanych przez przeciwników Bajtka na serwerze gry, wyznaczy minimalny czas potrzebny Bajtkowi aby stać się największą komórką w grze. Przypominamy, że komórka Bajtka zaczyna z masą równą 2.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba naturalna N ($1 \leq N \leq 200\,000$) określająca liczbę komórek przeciwników Bajtka. W drugim (ostatnim) wierszu wejścia znajduje się ciąg N liczb naturalnych T_i ($1 \leq T_i \leq 10^{18}$) pooddzielanych pojedynczymi odstępami. Są to masy komórek kolejnych przeciwników Bajtka.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć jedna nieujemna liczba całkowita – minimalny czas w sekundach potrzebny aby komórka Bajtka stała się największa na serwerze (przy założeniu, że pozostałe komórki nie będą zjadać innych).

Jeżeli osiągnięcie tego celu jest niemożliwe, należy zamiast tego wypisać tylko jedno słowo NIE.

Ocenianie

Możesz rozwiązać zadanie w kilku prostszych wariantach – niektóre grupy testów spełniają pewne dodatkowe ograniczenia. Poniższa tabela pokazuje, ile punktów otrzyma Twój program, jeśli przejdzie testy z takim ograniczeniem.

Dodatkowe ograniczenia	Liczba punktów
$N \leq 10$	20
$N \leq 20$	40
Dla każdego i istnieje takie całkowite j , że $T_i = 2^j$.	20

Przykłady

Wejście dla testu aga0a:

```
7
1 2 2 10 5 12 1
```

Wyjście dla testu aga0a:

```
4
```

Wyjaśnienie do przykładu: Wystarczy najpierw wchłonąć jedną komórkę o masie 1, później dwie komórki o masie 2 i wreszcie komórkę o masie 5. Wówczas komórka Bajtka będzie miała masę 12 i będzie ex-aequo największą komórką na serwerze.

Wejście dla testu aga0b:

```
7
3 4 5 3 6 4 3
```

Wyjście dla testu aga0b:

```
NIE
```

Wyjaśnienie do przykładu: Nie da się wchłonąć żadnej komórki.

Wejście dla testu aga0c:

```
8
1 2 4 8 16 32 64 128
```

Wyjście dla testu aga0c:

```
7
```

Wyjaśnienie do przykładu: Da się jedynie wchłaniać w kolejności z wejścia. Ostatniej komórki wchłonąć już nie trzeba.

Pozostałe testy przykładowe

- test aga0d: Test o maksymalnym rozmiarze $N = 200\,000$, liczby to $(1, 2, 2^2, 2^3, \dots, 2^{59}, 10^{18}, 10^{18}, \dots, 10^{18})$. Odpowiedź to 60.