**Michał Pilipczuk**
Treść zadania, Opracowanie

**Alan Kutniewski**
Program

**Dostępna pamięć: 32 MB.** *OI, Etap III, dzień drugi, 07.04.2011*

# Patyczki

*Mały Jaś dostał od babci i dziadka prezent na urodziny. Jest nim pudełko pełne patyczków różnej długości i różnych kolorów. Jaś zastanawia się, czy z pewnych trzech patyczków z zestawu da się zbudować trójkąt o wszystkich bokach różnych kolorów. Jasia interesują tylko trójkąty niezdegenerowane, czyli takie o dodatnim polu.*

## Wejście

*W pierwszym wierszu standardowego wejścia znajduje się jedna liczba całkowita $k$ ($3 \leqslant k \leqslant 50$) — jest to liczba różnych kolorów patyczków. Kolory numerujemy od 1 do $k$.*

*W kolejnych $k$ wierszach znajdują się opisy patyczków poszczególnych kolorów. W wierszu o numerze $i + 1$ znajdują się liczby całkowite pooddzielane pojedynczymi odstępami, opisujące patyczki koloru $i$. Pierwsza z tych liczb, $n_i$ ($1 \leqslant n_i \leqslant 1\,000\,000$), oznacza liczbę patyczków koloru $i$. Po niej następuje $n_i$ liczb całkowitych oznaczających długości patyczków. Są to liczby całkowite dodatnie nie większe niż 1 000 000 000. Łączna liczba wszystkich patyczków nie przekracza 1 000 000.*

*W testach wartych przynajmniej 30% punktów zachodzi dodatkowy warunek: sumaryczna liczba patyczków nie przekracza 250.*

## Wyjście

*Twój program powinien wypisać (w pierwszym i jedynym wierszu standardowego wyjścia):*

- *albo sześć liczb całkowitych pooddzielanych pojedynczymi odstępami, opisujących sposób zbudowania trójkąta o różnokolorowych bokach w następującym formacie: kolor i długość pierwszego patyczka, kolor i długość drugiego patyczka oraz kolor i długość trzeciego patyczka,*
- *albo słowo* `NIE`*, jeżeli takie trzy patyczki nie istnieją.*

*Jeżeli istnieje wiele trójek patyczków w różnych kolorach, z których można zbudować trójkąt, Twój program może wypisać dowolną z nich.*

## Przykład

*Dla danych wejściowych:*

```
4
1 42
2 6 9
3 8 4 8
1 12
```

*jednym z poprawnych wyników jest:*

```
3 8 4 12 2 9
```

| *natomiast dla danych wejściowych:* | *poprawnym wynikiem jest:* |
|---|---|
| `3`<br>`1 1`<br>`1 2`<br>`1 3` | `NIE` |

# Rozwiązanie

## Przypadek trzech kolorów

Rozważymy najpierw prostsze zadanie, w którym patyczki są dostępne jedynie w trzech kolorach, tzn. $k = 3$. Oznaczmy te kolory jako czerwony, niebieski i żółty. Przyjmijmy, że trójkąt, którego szukamy, ma najdłuższy bok czerwony, długości $c$; średni niebieski, długości $n$; oraz najkrótszy żółty, długości $z$. Wówczas jedyny warunek potrzebny i wystarczający do jego konstrukcji to $n + z > c$.

Załóżmy, że dobraliśmy już patyczek czerwony długości $c$. Zauważmy, że jako patyczek niebieski możemy bez straty ogólności dobrać *najdłuższy* patyczek tego koloru, który jest *nie dłuższy* niż $c$. Istotnie, jeśli dla jakiejś trójki długości $c$, $n$, $z$ spełniony jest warunek $n + z > c$, to jeśli zwiększymy $n$, będzie on tym bardziej spełniony. Podobnie, wybrawszy już patyczek niebieski długości $n$, z patyczków żółtych możemy dobrać najdłuższy nie dłuższy niż $n$.

Rozumowanie to prowadzi już do algorytmu. Wpierw sortujemy patyczki każdego koloru względem ich długości i ustawiamy je na trzech listach. Następnie mamy 6 możliwości na to, którego koloru patyczek ma być najdłuższy, którego średni, a którego najkrótszy. Sprawdzamy każdą z tych możliwości; dla ustalenia uwagi przyjmijmy tę, którą rozważaliśmy w poprzednich akapitach. Po kolei iterujemy przez patyczki czerwone, poczynając od najkrótszych. Na listach niebieskich oraz żółtych patyczków trzymamy dwa wskaźniki, wskazujące w każdym momencie na najlepsze możliwe dobory patyczków tych kolorów dla rozważanego patyczka czerwonego. Za każdym razem, gdy rozważamy kolejny patyczek czerwony, zwiększamy wskaźniki na pozostałych listach: przesuwamy wskaźnik na liście niebieskiej, by wskazywał na najdłuższy patyczek nie dłuższy niż czerwony, a potem przesuwamy wskaźnik na liście żółtej, by wskazywał na najdłuższy patyczek nie dłuższy niż niebieski. Następnie sprawdzamy, czy z wskazywanej trójki patyczków da się utworzyć trójkąt. Jeśli tak, to go wypisujemy i kończymy działanie algorytmu, w przeciwnym razie sprawdzamy dalej. Rozumowanie z poprzednich akapitów dowodzi, że jeśli istnieje co najmniej jedna trójka tworząca trójkąt, to którąś z nich znajdziemy.

Oznaczmy przez $N = \sum_{i=1}^{k} n_i$ łączną liczbę patyczków. Pierwsza faza opisanego algorytmu (sortowanie) działa w czasie $O(N \log N)$, zaś druga w czasie $O(N)$, gdyż dla każdej z sześciu kombinacji kolorów wskaźniki przesuwają się po listach jedynie wprzód. Stąd cała procedura ma złożoność $O(N \log N)$. Zauważmy, że bardzo prosto da się ją przerobić na, niestety zbyt wolny, algorytm dla większych wartości $k$. Po prostu sortujemy każdy z kolorów z osobna, a następnie na $k$ sposobów ustalamy kolor, z którego wybierzemy najdłuższy patyczek, na $k - 1$ — z którego wybierzemy średni, oraz na $k - 2$ — z którego wybierzemy najkrótszy. Dla każdej możliwości stosujemy znany już nam algorytm rozstrzygający, czy przy takim doborze możemy

zbudować szukany trójkąt. Metoda ma więc złożoność czasową $O(N \log N + k^3 N)$. Jej implementacja znajduje się w plikach `pats1.cpp` i `pats2.pas`.

## Rozwiązanie wzorcowe

Spróbujemy teraz poprawić algorytm tak, by również w ogólnym przypadku działał w czasie $O(N \log N)$, czyli niezależnie od liczby kolorów. Wyobraźmy sobie, że posortowaliśmy niemalejąco długości wszystkich patyczków naraz, i przy każdym patyczku zapamiętaliśmy, jaki ma kolor. Załóżmy, że istnieje różnokolorowy trójkąt, w którym najdłuższy bok stanowi patyczek $P_1$, długości $d_1$ i koloru $k_1$. Niech $P'_2$, $P'_3$ będą pozostałymi dwoma bokami tego trójkąta, odpowiednio długości $d'_2$, $d'_3$ ($d'_2 \geqslant d'_3$) oraz kolorów $k'_2$, $k'_3$. Wówczas wszystkie kolory $k_1$, $k'_2$ i $k'_3$ są różne oraz $d'_2 + d'_3 > d_1$. Niech dalej $d_2$ będzie długością najdłuższego patyczka $P_2$ nie dłuższego niż $d_1$, koloru innego niż $k_1$ (powiedzmy $k_2$). Ponadto, niech $d_3$ będzie długością najdłuższego patyczka $P_3$ nie dłuższego niż $d_2$, koloru innego niż $k_1$ oraz $k_2$ (powiedzmy $k_3$). Pokażemy, że wówczas patyczki $P_1$, $P_2$ i $P_3$ również tworzą poprawny trójkąt.

Wystarczy uzasadnić, że $d_2 \geqslant d'_2$ oraz $d_3 \geqslant d'_3$. Pierwsza nierówność jest oczywista, gdyż $k'_2 \neq k_1$, więc patyczek $P'_2$ jest brany pod uwagę przy znajdowaniu $P_2$. Aby dowieść drugiej, załóżmy najpierw, że $k'_2 \neq k_2$. Wówczas przy znajdowaniu $P_3$ patyczek $P'_2$ jest brany pod uwagę, gdyż jest nie dłuższy niż $d_2$ oraz innego koloru niż zarówno $P_1$, jak i $P_2$. Stąd $d_3 \geqslant d'_2$, więc też $d_3 \geqslant d'_3$. Teraz załóżmy, że jednak $k'_2 = k_2$. Ale wtedy z kolei patyczek $P'_3$ jest brany pod uwagę przy doborze $P_3$, gdyż jest innego koloru niż $P_1$ oraz $P_2$, więc w tym przypadku również $d_3 \geqslant d'_3$.

Dowiedziony fakt pozwala już wykonać stosowną modyfikację poprzedniego algorytmu. Jako potencjalne najdłuższe boki trójkąta rozważamy kolejne patyczki $P_1$ na posortowanej liście długości. Obserwacja z poprzedniego akapitu pozwala dla takiego $P_1$ sprawdzać tylko jedną parę patyczków jako dwa pozostałe boki: $P_2$, najdłuższy nie dłuższy od $P_1$ innego koloru niż $P_1$; oraz $P_3$, najdłuższy nie dłuższy od $P_2$, innego koloru niż $P_1$ i $P_2$. Po każdym kroku algorytmu pamiętamy takie trzy patyczki $P_1$, $P_2$, $P_3$ (lub informację, że danego patyczka nie ma). Powiedzmy, że rozważamy kolejny patyczek $P$. Wówczas aktualizujemy zapamiętane patyczki w następujący sposób:

- jeśli $P$ jest koloru innego niż $P_1$ i $P_2$, to $P_3 := P_2$, $P_2 := P_1$, $P_1 := P$;
- jeśli $P$ jest tego samego koloru co $P_2$, to $P_2 := P_1$, $P_1 := P$;
- jeśli $P$ jest tego samego koloru co $P_1$, to $P_1 := P$.

Oznacza to, że po wstępnym sortowaniu o złożoności czasowej $O(N \log N)$ możemy już w czasie liniowym sprawdzić, czy z którejś takiej trójki patyczków $P_1$, $P_2$, $P_3$ możemy skonstruować trójkąt, co jest równoważne istnieniu jakiejkolwiek trójki spełniającej warunki zadania.

Implementacje tego rozwiązania można znaleźć w plikach `pat2.cpp` oraz `pat3.pas`.

## Testy

Rozwiązania zawodników były sprawdzane z użyciem 13 zestawów składających się jednego, dwóch lub trzech testów.

| Nazwa | k | N | Opis |
|---|---|---|---|
| *pat1.in* | 3 | 15 | test losowy, odpowiedź pozytywna |
| *pat2.in* | 3 | 50 | test losowy, odpowiedź pozytywna |
| *pat3.in* | 4 | 150 | test losowy, odpowiedź pozytywna |
| *pat4a.in* | 5 | 250 | test losowy, odpowiedź pozytywna |
| *pat4b.in* | 4 | 180 | test strukturalny, odpowiedź negatywna |
| *pat5a.in* | 5 | 485 | test strukturalny, odpowiedź negatywna |
| *pat5b.in* | 7 | 32 344 | test strukturalny, jednoznaczna odpowiedź pozytywna |
| *pat6a.in* | 5 | 1 370 | test strukturalny, odpowiedź negatywna |
| *pat6b.in* | 7 | 42 695 | test strukturalny, jednoznaczna odpowiedź pozytywna |
| *pat7a.in* | 5 | 6 881 | test strukturalny, odpowiedź negatywna |
| *pat7b.in* | 6 | 103 077 | test strukturalny, jednoznaczna odpowiedź pozytywna |
| *pat8a.in* | 11 | 25 738 | test strukturalny, odpowiedź negatywna |
| *pat8b.in* | 9 | 155 756 | test strukturalny, jednoznaczna odpowiedź pozytywna |
| *pat8c.in* | 10 | 101 702 | test strukturalny, odpowiedź negatywna |
| *pat9a.in* | 14 | 68 830 | test strukturalny, odpowiedź negatywna |
| *pat9b.in* | 9 | 263 487 | test strukturalny, jednoznaczna odpowiedź pozytywna |
| *pat9c.in* | 11 | 153 583 | test strukturalny, odpowiedź negatywna |
| *pat10a.in* | 28 | 500 000 | test losowy, odpowiedź pozytywna |
| *pat10b.in* | 9 | 323 145 | test strukturalny, jednoznaczna odpowiedź pozytywna |
| *pat10c.in* | 9 | 236 090 | test strukturalny, odpowiedź negatywna |
| *pat11a.in* | 35 | 500 000 | test losowy, odpowiedź pozytywna |
| *pat11b.in* | 11 | 357 247 | test strukturalny, jednoznaczna odpowiedź pozytywna |
| *pat11c.in* | 12 | 305 079 | test strukturalny, odpowiedź negatywna |
| *pat12a.in* | 45 | $10^6$ | test losowy, odpowiedź pozytywna |
| *pat12b.in* | 10 | 406 820 | test strukturalny, jednoznaczna odpowiedź pozytywna |
| *pat12c.in* | 9 | 441 183 | test strukturalny, odpowiedź negatywna |
| *pat13a.in* | 50 | $10^6$ | test losowy, odpowiedź pozytywna |
| *pat13b.in* | 15 | 474 711 | test strukturalny, jednoznaczna odpowiedź pozytywna |
| *pat13c.in* | 12 | 531 982 | test strukturalny, odpowiedź negatywna |

# XXIII Międzynarodowa Olimpiada Informatyczna,

*Pattaya, Tajlandia 2011*