

# Problem 24: Trzech do towarzystwa

Punkty: 70

Autor: Louis Ronat, Denver, Kolorado, Stany Zjednoczone

## Wprowadzenie

Programy komputerowe są skomplikowane. Jednak analizowanie, *jak* bardzo skomplikowane są programy komputerowe - ustalanie ich złożoności obliczeniowej - to podstawowy problem nauk komputerowych. Istnieją różne klasy złożoności obliczeniowej, a jednym z największych pytań bez odpowiedzi w analizie złożoności dotyczy klasy znanej jako problemy NP. Problem NP to taki, w którym uważa się, że algorytm rozwiązujący problem potrzebuje ilości czasu, która rośnie szybciej niż wielomian porównywalny z rozmiarem danych wejściowych. To jest, jeśli  $a$  jest stałą, a  $n$  to rozmiar danych wejściowych, to czas potrzebny na rozwiązanie problemu rośnie szybciej niż wartość  $n^a$ . Wiele najtrudniejszych problemów w zawodach Code Quest Lockheed Martin należy do tej kategorii.

Choć problemy te są niezwykle skomplikowane, weryfikacja prawdziwości danego rozwiązania nie jest równie trudna. Problemy NP charakteryzują się tym, że weryfikację podanego rozwiązania można przeprowadzić w czasie wielomianowym. Dziś będziecie weryfikować taki problem NP, znany jako 3-SAT lub problem spełnialności 3.

## Opis problemu

Problemy 3-SAT zaczynają się od zdania logicznego pogrupowanego w określony sposób. Zdanie takie zawiera co najmniej dwie „trójki” danych logicznych połączonych operatorem logicznym OR. Poniżej przedstawiono taką trójkę:

**A or B or C**

Następnie takie trójki łączy się operatorem logicznym AND, by stworzyć pełne zdanie. Na przykład:

**(A or B or C) and (!D or !E or !F)**

Należy pamiętać, że znak ! oznacza tutaj zaprzeczenie; !D jest prawdziwe, gdy D jest fałszywe.

Rozwiązanie problemu 3-SAT wymaga udzielenia odpowiedzi na pytanie: „Czy to zdanie będzie kiedykolwiek prawdziwe?” Podane powyżej zdanie będzie prawdziwe, jeśli co najmniej jedna z wartości A, B lub C jest prawdziwa i co najmniej jedna z wartości D, E lub F jest fałszywa. O ile pewne kombinacje dopuszczalnych wartości wejściowych spełniają podane kryteria, to odpowiedź na pytanie brzmi tak; jeśli nie ma takiej kombinacji wartości wejściowych, to odpowiedź brzmi nie.

Przykładowo, rozważmy sytuację, gdzie takie zdanie umieszcza się w instrukcji warunkowej IF w wybranym przez was języku programowania. Odpowiedź na problem 3-SAT brzmi „tak”, o ile istnieje

sposób wykonania kodu umieszczonego wewnątrz takiego bloku instrukcji IF. Jeśli nie da się tego kodu wykonać, odpowiedź brzmi „nie”.

3-SAT może mieć więcej niż dwie trójki; w każdym przypadku elementy takiej trójki są łączone operatorem logicznym OR, a trójki są łączone ze sobą w zdania operatorem AND.

Wasz zespół musi napisać program, który jest w stanie zbudować zdanie 3-SAT i przetestować je z wykorzystaniem kilku danych wejściowych w celu ustalenia, czy problem jest rozwiązywalny. Uważajcie jednak; dane wejściowe o charakterze logicznym są podawane w kolejności alfabetycznej, zatem musicie upewnić się, że zostały prawidłowo zmapowane na zmienne w trójkach.

## Przykładowe dane wejściowe

Pierwszy wiersz danych wejściowych programu, otrzymanych przez standardowy kanał wejściowy, będzie zawierał dodatnią liczbę całkowitą oznaczającą liczbę przypadków testowych. Każdy przypadek testowy będzie zawierał:

- Wiersz zawierający dwie dodatnie liczby całkowite oddzielone spacjami:
  - Pierwsza liczba całkowita, T, reprezentująca liczbę trójek w zdaniu 3-SAT. T będzie większa od 2 lub jej równa.
  - Druga liczba całkowita, I, reprezentuje liczbę danych wejściowych służących do przetestowania zdania końcowego.
- T wierszy zawierających trzy wartości wejściowe, oddzielone spacjami, odpowiadające trójkom tworzącym zdanie 3-SAT. Każdy wiersz reprezentuje pojedynczą trójkę, a każda wartość wejściowa składa się z jednej wielkiej litery. Litery mogą być poprzedzone wykrzyknikiem (!), który oznacza, że w zdaniu 3-SAT należy zamieścić zaprzeczenie takiej wartości wejściowej. W każdym przypadku testowym użyte litery zaczynają się od A i nie pomija się żadnych liter, jednak litery nie muszą być przedstawiane w kolejności alfabetycznej.
- I wierszy, z których każdy zawiera 3T zer (0) lub jedynek (1) oddzielonych spacjami, odpowiadających wartościom wejściowym, które mają posłużyć do przetestowania zdania 3-SAT. 0 oznacza fałsz, a 1 prawdę. Wartości są prezentowane w kolejności alfabetycznej; pierwsza odpowiada wartości A, druga wartości B itd.

```
1
2 4
A B C
!D !E !F
1 0 0 1 1 0
1 1 1 1 1 1
0 0 0 0 0 0
1 1 1 0 0 0
```

## Przykładowe dane wyjściowe

W każdym przypadku testowym wasz program musi wyświetlić jeden wiersz dla każdego zestawu testowanych wartości, podając słowo „TRUE”, jeśli z zadanymi wartościami zdanie 3-SAT zwraca prawdę lub słowo „FALSE” w przeciwnym wypadku.

```
TRUE
FALSE
FALSE
TRUE
```