



POTYCZKI ALGORYTMICZNE
ADB POLSKA & MIMUW

Zadanie: FOR Formuły

Tura 4, plik źródłowy for.*, dostępna pamięć 32 MB

04 marca 2005

Jak mawiają górale, są trzy rodzaje prawdy: święta prawda, tyż prawda i ... prawda. Najnowsze badania bajtockich logików potwierdzają to. Badają oni wyrażenia logiczne (formuły zdaniowe) o uproszczonej postaci nazywanej klauzulową. Postać klauzulowa wygląda następująco:

- Formuły budujemy używając *zmiennych logicznych* x_1, x_2, \dots, x_n . Zmienne te mogą przyjmować wartości prawda lub fałsz.
- *Literał*, to zmienna logiczna lub jej negacja, x_i lub $\neg x_i$ (dla $1 \leq i \leq n$).
- *Klauzula*, to koniunkcja literałów, np. $x_1 \wedge \neg x_2 \wedge x_3$.
- *Formuła*, to alternatywa klauzul, np. $(x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_3) \vee (\neg x_3)$.

Wartość formuły zależy od konkretnych wartości zmiennych. Wartości zmiennych określa się za pomocą funkcji nazywanej *wartościowaniem*, postaci $f: \{1, 2, \dots, n\} \rightarrow \{\text{prawda}, \text{fałsz}\}$, gdzie $f(i)$ jest wartością nadawaną zmiennej x_i . Wszystkich możliwych wartościowań jest 2^n . Dla konkretnego wartościowania, wartością danej formuły jest albo prawda albo fałsz.

Mówimy, że formuła jest prawdziwa, jeśli dla każdego wartościowania jej wartością jest prawda. Powiemy, że formuła jest fałszywa, jeżeli dla każdego wartościowania jej wartością jest fałsz. Często formuła nie jest ani prawdziwa, ani fałszywa, ale jej wartość zależy faktycznie od danego wartościowania. Prawdziwość formuły możemy określić jako ułamek (z zakresu od 0 do 1):

$$\frac{\text{liczba wartościowań, dla których wartością formuły jest prawda}}{2^n}$$

Liczba 1 odpowiada formułom prawdziwym, a 0 fałszywym.

Przykład

Formuła:

$$(x_1 \wedge \neg x_1) \vee (\neg x_2 \wedge \neg x_3) \vee (x_3 \wedge x_2)$$

jest prawdziwa dla czterech spośród ośmiu możliwych wartościowań. Tak więc jest ona prawdziwa w połowie.

Zadanie

Dysponujesz 11 zestawami danych umieszczonych w dziale Przydatne zasoby. Każdy zestaw jest zapisany w osobnym pliku: `fork.in`, gdzie k jest numerem zestawu ($0 \leq k \leq 10$). Plik `for0.in` zawiera dane zgodne z powyższym przykładem. Napisz program, który na wejściu otrzymuje dokładnie jedną liczbę całkowitą k ($0 \leq k \leq 10$) i wypisuje dokładnie jeden wiersz wyjścia, zawierający przybliżoną prawdziwość formuły podanej w k -tym zestawie danych. Prawdziwość powinna być wypisana w postaci ułamka dziesiętnego, obliczonego ze względną dokładnością 3%. To znaczy, jeśli prawdziwość formuły to p , a jej obliczone przybliżenie to p' , to musi zachodzić $|p' - p| \leq 0.03p$.

Opis pojedynczego pliku wejściowego

W pierwszym wierszu wejścia znajdują się dwie dodatnie liczby całkowite n i m , $1 \leq n \leq 100$, $1 \leq m \leq 100$, oddzielone pojedynczym odstępem. Zmienne logiczne, które mogą pojawiać się w formule to x_1, x_2, \dots, x_n . Formuła składa się z m klauzul, opisanych w kolejnych m wierszach, po jednej w wierszu. Każdy z tych wierszy zawiera liczby całkowite pooddzielane pojedynczymi odstępami. Pierwsza z tych liczb, l , $1 \leq l \leq n$, to liczba literałów tworzących klauzulę. Dalej w wierszu znajduje się l liczb całkowitych reprezentujących kolejne literały klauzuli. Są to liczby różne od zera, z zakresu od $-n$ do n . Liczba i (dla $1 \leq i \leq n$) reprezentuje x_i , a liczba $-i$ reprezentuje $\neg x_i$.

Przykład

Dla danych wejściowych:

0

poprawnym wynikiem jest:

0.5

Uwaga

Akceptowanymi odpowiedziami do przykładowego testu są również: 0.485, 0.498, 0.51, 0.515