

Szeregowanie bloków bazowych

Jedną z technik optymalizacji kodu assemblerowego generowanego przez kompilator języka wysokiego poziomu jest szeregowanie bloków bazowych w celu minimalizacji liczby skoków. Blok bazowy jest ciągiem kolejnych rozkazów assemblerowych spełniającym następujące warunki:

- żaden rozkaz, z wyjątkiem ostatniego w bloku, nie jest rozkazem skoku,
- żaden rozkaz, z wyjątkiem pierwszego w bloku, nie jest pierwszym rozkazem wykonywanym po rozkazie skoku (także z innego bloku bazowego).

Technika optymalizacji, której dotyczy to zadanie, korzysta z faktu, że gdy dwa bloki bazowe ułożone będą w programie kolejno, to łącząca je instrukcja skoku będzie mogła zostać pominięta.

Twoim zadaniem jest napisanie programu, który bazując na informacji zwrotnej profilera zawierającej dane o częstościach przejść między blokami bazowymi, obliczy ile rozkazów skoków byłoby wykonanych, gdyby w badanym programie bloki bazowe były uszeregowane w sposób optymalny.

Dane wejściowe:

W pierwszym wierszu danych wejściowych znajduje się pojedyncza liczba całkowita C oznaczająca liczbę zestawów danych. Dalej znajdują się kolejne zestawy danych.

Pierwszy wiersz zestawu danych zawiera pojedynczą liczbę całkowitą N oznaczającą liczbę bloków bazowych w badanym programie. W kolejnych N wierszach znajdują się opisy przejść wychodzących z kolejnych bloków bazowych. Na początku każdego z tych wierszy znajduje się nazwa początkowego bloku bazowego, będąca jedną z N pierwszych małych liter alfabetu angielskiego (litery od 's' do 'z' są zarezerwowane do innych celów, więc nie pojawiają się w danych wejściowych). Dalej występuje znak strzałki „->”, a następnie liczba M ($0 \leq M \leq N$) oznaczająca liczbę bloków bazowych, do których prowadzą skoki. Pozostała część wiersza zawiera M par, w których pierwszym elementem jest nazwa docelowego bloku bazowego, a drugim liczba przejść odnotowanych przez program profilujący. Wszystkie te elementy pooddzielane są spacjami.

Uwagi:

- Jedynym ograniczeniem na kolejność bloków bazowych jest to, że blok oznaczony literą 'a' („blok początkowy”) musi wystąpić w programie jako pierwszy, a blok oznaczony najwyższą literą („blok końcowy”) musi wystąpić jako ostatni.
- Dla wszystkich bloków z wyjątkiem bloków brzegowych (tj. początkowego i końcowego) zachodzi równość (prawo Kirchhoffa):

$$\text{liczba przejść dochodzących (wejść)} = \text{liczba przejść wychodzących (wyjść)}.$$

- W bloku początkowym liczba wyjść jest o 1 większa od liczby wejść.
- W bloku końcowym liczba wyjść jest o 1 mniejsza niż liczba wejść.

Dane wyjściowe:

Dane wyjściowe zawierają C wierszy (po jednym dla każdego zestawu danych), które zawierają wyniki dla kolejnych zestawów danych, w kolejności identycznej jak w danych wejściowych. Odpowiedzią dla pojedynczego zestawu danych jest pojedyncza liczba całkowita oznaczająca całkowitą liczbę skoków, jakie byłyby wykonane, gdyby bloki bazowe były uszeregowane w sposób optymalny.

Przykładowe dane wejściowe:

```
2
8
a -> 2 b 5 f 6
b -> 2 c 3 d 2
c -> 1 d 3
d -> 1 f 6
e -> 1 d 1
f -> 1 g 12
g -> 3 a 10 e 1 h 1
h -> 0
7
a -> 2 b 5 f 6
b -> 2 c 3 d 2
c -> 1 d 3
d -> 1 f 6
e -> 1 d 1
f -> 1 g 12
g -> 2 a 10 e 1
```

Przykładowe dane wyjściowe:

```
20
22
```

Wyjaśnienie:

Dla przykładowych danych optymalne sekwencje bloków bazowych są następujące:

```
abcdfgeh
abcedfg
```