

Zadanie «Dom Bankowy» (dom)

W Domu Bankowym Gustawa Kramera dane kont klientów są przechowywane w drzewie poszukiwań binarnych uporządkowanym według numerów kont. Dla uproszczenia przyjmujemy, że numery kont są kolejnymi liczbami naturalnymi począwszy od 1. Numery kont nie powtarzają się. Niektórzy klienci często dokonują operacji bankowych na swoim koncie, inni z kolei traktują konta jako lokaty na czarną godzinę i bardzo rzadko wpłacają lub wypłacają pieniądze. Prezes Kramer zlecił optymalizację przechowywania danych w drzewie poszukiwań binarnych. Główny analityk domu bankowego Stawiski zaproponował, żeby konta używane częściej przesunąć bliżej korzenia drzewa, z kolei konta używane rzadziej – w kierunku liści drzewa. Wtedy konta używane częściej będą szybciej odszukiwane.

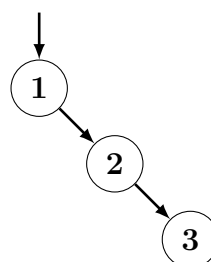
Analityk Stawiski przygotował następujący przykład dla trzech klientów. Konta tych klientów mają numery 1, 2 i 3. Liczba operacji na koncie pierwszego wynosi $k_1 = 2$, klienta drugiego – $k_2 = 25$, a klienta trzeciego – $k_3 = 12$. Pojedynczy czas dostępu do konta jest równy wysokości konta w drzewie (przyjmujemy, że korzeń ma wysokość 1). Całkowity czas dostępu do kont wynosi $t = k_1h_1 + k_2h_2 + k_3h_3 = 2 \cdot 1 + 25 \cdot 2 + 12 \cdot 3 = 88$.

wysokość

1

2

3

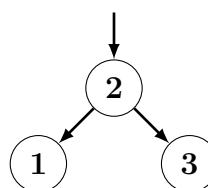


Wystarczy jednak przeorganizować drzewo (zachowując własności drzewa poszukiwań binarnych):

wysokość

1

2



Teraz czas wynosi $t = k_1h_1 + k_2h_2 + k_3h_3 = 2 \cdot 2 + 25 \cdot 1 + 12 \cdot 2 = 53$.

Prezes Kramer jest zachwycony pomysłem Stawiskiego. Jednak Stawiski nie jest informatykiem i nie potrafi przekształcić tak drzewa poszukiwań binarnych, żeby sumaryczny

czas dostępu do kont

$$t = \sum_{i=1}^n k_i h_i,$$

(gdzie k_i to liczba operacji na i -tym koncie, a h_i to wysokość i -tego konta w drzewie) był minimalny.

Specyfikacja wejścia

Wejście zawiera 15 zestawów danych testowych. Każdy zestaw zawarty jest w jednej linii i składa się z (nie więcej niż 6000) liczb naturalnych (nieprzekraczających wartości 5000) reprezentujących liczbę operacji na kontach o numerach $1, 2, \dots, n$, gdzie $n < 5000$.

Specyfikacja wyjścia

Program powinien wypisać dla każdego zestawu jedną liczbę, będącą minimalnym całkowitym czasem dostępu do kont modulo 999999937. Plik wyjściowy zawiera dokładnie tyle samo linii, ile plik wejściowy.

Przykładowe wejście

```
10
10 1
1 10 3 2 1 9
```

Przykładowe wyjście

```
10
12
52
```

W ostatnim przykładzie liczby operacji na kontach $1, 2, \dots, 6$ wynoszą odpowiednio $k_1 = 1$, $k_2 = 10$, $k_3 = 3$, $k_4 = 2$, $k_5 = 1$, $k_6 = 9$. Drzewo o minimalnym całkowitym czasie dostępu ma postać:

wysokość

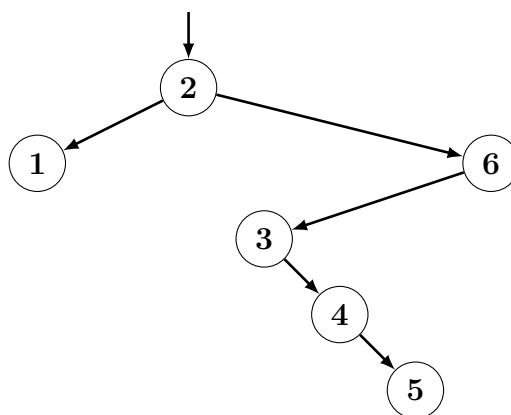
1

2

3

4

5



Czas całkowity dostępu do kont wynosi: $t = \sum_{i=1}^6 k_i h_i = 1 \cdot 2 + 10 \cdot 1 + 3 \cdot 3 + 2 \cdot 4 + 1 \cdot 5 + 9 \cdot 2 = 52 \equiv 52 \pmod{999999937}$.

Ustalenia techniczne

1. Rozwiązaniem zadania są:

- program konsolowy napisany w języku C/C++ – Kod źródłowy programu powinien być zawarty wyłącznie w jednym pliku o nazwie `dom.c` (dla języka C) lub `dom.cpp` (dla języka C++). W pierwszej linii pliku źródłowego należy umieścić w komentarzu *indywidualny kod uczestnika* (IKU). Nie jest dopuszczalne umieszczanie w kodzie jakichkolwiek innych danych umożliwiających zidentyfikowanie uczestnika (także we właściwościach pliku).
- plik wyjściowy `dom.out` wypracowany przez program dla danych testowych dostarczonych wraz z treścią zadania – Plik musi być nazwany zgodnie z niżej umieszczonym nazewnictwem. Plik musi być zgodny ze specyfikacją wyjścia.

Wszystkie powyższe pliki należy spakować do pliku `IKU-dom.zip`, gdzie IKU jest *indywidualnym kodem uczestnika*.

2. Program powinien odczytywać dane wejściowe z pliku o nazwie podanej w treści zadania, a wynik należy zapisać też do pliku, którego nazwa jest podana w treści zadania.
3. Należy przyjąć, że dane wejściowe mają poprawny format (opisany w treści zadania). Plik wyjściowy powinien mieć format opisany w treści zadania.
4. W programach można korzystać wyłącznie ze standardowych bibliotek języka C/C++.

5. W programach nie można korzystać z rozwiązań i mechanizmów nieprzenośnych (np. zależnych od systemu operacyjnego).
6. Programy nie mogą:
 - tworzyć nowych procesów lub wątków,
 - uruchamiać innych programów,
 - używać funkcji sieciowych (np. `socket`, `send` itp.),
 - oczekiwać na interakcję użytkownika.
7. Zadanie należy przesłać przez stronę konkursu «Złoty Indeks» Platformy Zdalnej Edukacji korzystając z łącza do przesyłania rozwiązań zadania «dom».
8. Zadanie jest oceniane w skali 0-15 punktów.

Nazewnictwo plików

Dane wejściowe znajdują się w pliku o nazwie `dom.in`, plik ten zawiera wiele zestawów danych wejściowych. Dane wyjściowe (rozwiązania dla wszystkich zestawów danych wejściowych) powinny zostać zapisane do jednego pliku wyjściowego o nazwie `dom.out`.