Jakub Wojtaszczyk
Treść zadania, Opracowanie

Marcin Andrychowicz
Program

Dostępna pamięć: 128 MB.

OI, etap II, dzień pierwszy, 13.02.2013

Inspektor

Inspektor Bajtazar bada zbrodnię, która wydarzyła się w biurze informatyków. Próbuje teraz ustalić chronologię zdarzeń. Niestety, informatycy są jednostkami stosunkowo roztrzepanymi. Żaden z nich nie udziela rozsądnych informacji, mówią raczej: „No, kiedy zajrzałem na serwer o 14:42, to zobaczyłem, że zalogowanych w pracy jest pięciu innych informatyków”.

Wiadomo, że każdy z informatyków tego dnia przyszedł do biura, spędził w nim trochę czasu i wyszedł. Żaden informatyk nie opuszczał biura pomiędzy swoim przyjściem a wyjściem i nie pojawiał się w biurze poza tym odcinkiem czasu.

Bajtazar nie jest pewien, czy może polegać na zeznaniach informatyków. Chce się dowiedzieć, czy w ogóle możliwe jest, żeby wszyscy mówili prawdę. Poprosił Cię o pomoc.

Wejście

W pierwszym wierszu standardowego wejścia znajduje się liczba całkowita z ($1 \leqslant z \leqslant 50$), oznaczająca liczbę zestawów danych. W kolejnych wierszach znajdują się opisy z zestawów danych.

W pierwszym wierszu opisu znajdują się dwie liczby całkowite n i m oddzielone pojedynczym odstępem ($1 \leqslant n, m \leqslant 100\,000$). Oznaczają one odpowiednio liczbę informatyków w biurze i liczbę zeznań zebranych przez Bajtazara. Informatycy są ponumerowani od 1 do n.

W każdym z kolejnych m wierszy opisane jest jedno zeznanie. Każdy z tych wierszy zawiera trzy liczby całkowite t, j oraz i pooddzielane pojedynczymi odstępami ($1 \leqslant t \leqslant m$, $1 \leqslant j \leqslant n$, $0 \leqslant i < n$). Oznaczają one, że informatyk numer j zeznał, iż w chwili t był w biurze i oprócz niego było tam jeszcze i innych informatyków. Większa liczba oznacza późniejszą chwilę. Przyjmujemy, że informatycy przychodzili do pracy i wychodzili z pracy w czasie przed, po lub pomiędzy chwilami, których dotyczą zeznania.

W testach wartych łącznie 7% punktów zachodzi dodatkowy warunek $n, m \leqslant 5$. Ponadto w testach wartych łącznie 28% punktów zachodzi dodatkowy warunek $n, m \leqslant 101$.

Wyjście

Dla każdego zestawu danych Twój program powinien wypisać na standardowe wyjście jedną dodatnią liczbę całkowitą. Wypisanie liczby k ($1 \leqslant k \leqslant m$) oznacza, że pierwsze k zeznań podanych na wejściu dla danego zestawu może być prawdziwych, natomiast pierwsze $k+1$ zeznań nie może być prawdziwych. W szczególności, jeśli $k = m$, to wszystkie zeznania podane na wejściu mogą być prawdziwe.

Przykład

Dla danych wejściowych:

```
2
3 5
1 1 1
1 2 1
2 3 1
4 1 1
4 2 1
3 3
3 3 0
2 2 0
1 1 0
```

poprawnym wynikiem jest:

```
4
3
```

Wyjaśnienie do przykładu: *W pierwszym zestawie danych nie wszystkie zeznania mogą być prawdziwe. Informatycy 1 i 2 zeznali, że byli w biurze przynajmniej od chwili 1 do chwili 4, natomiast informatyk 3 zeznał, że w chwili 2 w biurze była tylko jedna osoba oprócz niego. Jeśli odrzucimy ostatnie zeznanie, to pozostałe mogą już być prawdziwe. Wystarczy, żeby informatyk 2 wyszedł z biura między chwilą 1 i 2.*

W drugim zestawie danych wszystkie złożone zeznania mogą być prawdziwe.

Testy „ocen":

`1ocen`: $z = 1$, $n = 5$, $m = 5$, *wszystkie zeznania są ze sobą zgodne, żaden informatyk nie widział się z żadnym innym.*

`2ocen`: $z = 50$, *w każdym zestawie danych* $n = 101$, $m = 101$. *W zestawach o nieparzystych indeksach wszystkie zeznania są ze sobą zgodne i każdy informatyk widział się z każdym innym. Zestawy o parzystych indeksach różnią się tym, że zeznania dotyczące jednego momentu są składane przez większą liczbę informatyków, niż wynika to z owych zeznań.*

`3ocen`: $z = 1$, $n = 100\,000$, $m = 50\,000$, *wszystkie zeznania poza jednym wyglądają tak samo, zaś to jedno pozostałe jest z nimi sprzeczne.*

Rozwiązanie

Zacznijmy od zrozumienia, jakie informacje są zawarte w pojedynczym zeznaniu. Po pierwsze, jeśli informatyk j zeznaje, że w chwili t było w biurze i osób oprócz niego, to wiemy, że w chwili t łącznie było $i + 1$ osób. Po drugie, wiemy, że informatyk j był w biurze w chwili t. Z treści zadania wynika, że każdy informatyk był w pracy w pewnym spójnym przedziale czasu.

W zadaniu pytani jesteśmy o to, jak wiele początkowych zeznań może być niesprzeczne. Zauważmy, że jeśli początkowe k zeznań jest niesprzeczne, to i dowolna mniejsza początkowa liczba zeznań jest niesprzeczna – a zatem do znalezienia największego k takiego, że pierwsze k zeznań jest niesprzeczne, będziemy mogli zastosować *wyszukiwanie binarne*. Do tego wystarczy nam umieć odpowiedzieć na pytanie „czy dany zbiór zeznań jest niesprzeczny".

Nierozróżnialni informatycy

Zaczniemy od prostszego zadania, w którym będziemy ignorować informacje drugiego typu (mówiące, że dany informatyk był w biurze w chwili, której dotyczy jego zeznanie). Innymi słowy, spróbujemy rozwiązać zadanie, w którym będziemy mieli zbiór zdarzeń postaci „w pracy jest i informatyków". Każde zdarzenie będzie reprezentowane przez parę (t_l, i_l), która oznacza, że w chwili t_l w biurze było i_l informatyków.

Po pierwsze, jeśli dla pewnej chwili t mamy więcej niż jedno zeznanie, to wszystkie te zeznania muszą się zgadzać. Ten warunek łatwo można sprawdzić na początku, sortując zeznania po czasie i sprawdzając dla sąsiednich zeznań, czy nie mają równych czasów a różnych liczb informatyków.

Po drugie, mogą nam się „skończyć" informatycy. Przykładowo, jeśli informatyków jest czterech i wiemy, że w chwilach $t = 1$ i $t = 3$ było ich w biurze po trzech, zaś w $t = 2$ był tylko jeden, to zeznania są sprzeczne. Wiemy z jednej strony, że dwóch informatyków wyszło z biura między $t = 1$ a $t = 2$, a z drugiej strony w $t = 3$ powinno być ich w biurze trzech – a przecież informatyk po opuszczeniu biura do niego nie wraca.

Taki problem też łatwo wykryć. Możemy przeglądać zeznania chronologicznie i pamiętać, ilu informatyków ostatnio było widzianych w biurze i ilu wiemy na pewno, że już wyszło. Suma tych dwóch liczb nie może przekroczyć łącznej liczby informatyków. Gdy rozważamy dane zeznanie, jeśli wymaga ono zwiększenia liczby informatyków w biurze, to zwiększamy ją, a jeśli wymaga zmniejszenia tej liczby, to odejmujemy odpowiednią wartość od liczby informatyków przebywających obecnie w biurze i dodajemy tę samą wartość do liczby informatyków, którzy już biuro opuścili. Uzasadnimy teraz, że jeśli przejrzymy wszystkie zdarzenia i nigdy nie przekroczymy liczby informatyków, to ich zeznania faktycznie są niesprzeczne.

Rozwiązaniem nazwiemy przypisanie każdemu z informatyków godziny przyjścia i wyjścia z biura. Powiemy, że rozwiązanie jest *poprawne*, jeśli zgadza się ze wszystkimi zeznaniami, które mamy do dyspozycji (w tej części oznacza to tyle, że w chwili dotyczącej dowolnego zeznania liczba informatyków się zgadza; dalej będziemy również sprawdzać, że każdy informatyk jest obecny w każdej chwili, dla której składa zeznanie).

Zauważmy teraz, że algorytm podany powyżej odpowiada zachłannej konstrukcji rozwiązania – zakładamy, że pomiędzy dwoma kolejnymi zeznaniami t_1 i t_2, jeśli $i_1 > i_2$, to wychodzi $i_1 - i_2$ informatyków, a jeśli $i_1 < i_2$, to przychodzi $i_2 - i_1$ informatyków. Zauważmy, że jest to jedyne możliwe poprawne rozwiązanie, jeśli tylko założymy, że w żadnej przerwie pomiędzy zeznaniami nie zachodzi sytuacja, w której pewien informatyk wychodzi z biura, a inny informatyk przychodzi do biura.

Załóżmy zatem na chwilę, że mamy poprawne rozwiązanie, w którym pomiędzy pewnymi kolejnymi zeznaniami t_1 i t_2 wychodzi pewien informatyk A, zaś przychodzi informatyk B. Nazwiemy taką sytuację *zmianą warty*. Rozważmy dowolną zmianę warty i zmodyfikujmy nasze rozwiązanie, przedłużając obecność w pracy informatyka A aż do momentu, w którym pierwotnie B wychodził z pracy, i przyjmując, że B

do pracy nie przyszedł w ogóle. W ten sposób otrzymamy rozwiązanie, w którym w pracy jest o jeden mniej informatyk i w chwili dotyczącej każdego zeznania liczba informatyków jest taka sama jak w oryginalnym rozwiązaniu (a zatem nowe rozwiązanie jest również poprawne). Powtarzajmy tę procedurę dopóty, dopóki w rozwiązaniu będą jakieś zmiany warty. Ponieważ każda modyfikacja powoduje, że liczba informatyków, którzy przyszli do pracy, maleje o jeden, tak więc w końcu musimy osiągnąć poprawne rozwiązanie, w którym nie ma żadnej zmiany warty. Zatem, o ile istnieje jakiekolwiek poprawne rozwiązanie, istnieje też rozwiązanie bez zmian warty – czyli takie, które zostanie znalezione przez nasz algorytm.

Rozróżnialni informatycy

Spójrzmy, jak rozróżnianie informatyków, spowodowane przez informacje drugiego typu, komplikuje nam życie. Przykładowo, jeśli informatyków jest trzech i złożyli oni pięć zeznań, z których wynika, że w chwilach $t = 1, 2, 4, 5$ w biurze jest po jednym informatyku, zaś w chwili $t = 3$ są wszyscy trzej, to ten układ zeznań jest niesprzeczny. Ale jeśli wiemy, że zeznanie dla chwili $t = 1$ składał informatyk A, zaś dla $t = 2$ – informatyk B, to zeznania już są sprzeczne. Istotnie, informatyk A musiał wyjść z biura przed chwilą $t = 2$, a zatem nie mógł być obecny w chwili $t = 3$.

Gdyby informatyków było niewielu, moglibyśmy pokusić się o rozwiązanie oparte o programowanie dynamiczne, w którym pamiętalibyśmy, jaki zbiór informatyków obecnie jest w biurze. Takie rozwiązanie może jednak zadziałać tylko dla bardzo małych n ($n = 30$ będzie już zbyt duże). Niestety, w naszym zadaniu przyszło nam rozwiązywać zagadkę zbrodni w dużej korporacji informatycznej, a nie w świeżo założonej firmie, a więc musimy wymyślić coś sprytniejszego. Spróbujmy zatem znaleźć rozwiązanie oparte na podobnej argumentacji co poprzednio, które umożliwi nam stwierdzenie, kto z biura wychodzi, a kto do niego przychodzi.

Zmiany warty

W tej wersji zadania poprawne rozwiązanie może wymagać zmiany warty. Przykładowo, jeśli informatyk A zeznał, że w chwili t_1 był w pracy sam, a informatyk B zeznał, że w chwili t_2 był w pracy sam, to pomiędzy chwilami t_1 a t_2 będzie musiała nastąpić zmiana warty. Zauważmy, że ta zmiana warty jest wymuszona tym, że B *musiał* przyjść do pracy, żeby zeznawać. Dodajmy zatem do zbioru rozważanych zdarzeń zdarzenia postaci „informatyk X zeznaje po raz pierwszy". Te zdarzenia będziemy rozpatrywali przed zdarzeniami dotyczącymi liczby informatyków w pracy. W momencie takiego zdarzenia, jeśli X jeszcze nie był w pracy, to dodajemy go do zbioru informatyków, którzy już przyszli.

Analogią faktu stosowanego w poprzednim rozwiązaniu jest następujące stwierdzenie. Jeśli istnieje jakiekolwiek poprawne rozwiązanie, to istnieje takie, w którym:

- Bezpośrednio przed zdarzeniem „informatyk X zeznaje po raz pierwszy" być może do pracy nie przyszedł nikt, jeśli X w pracy już był, lub tylko X, jeśli X w pracy nie było. Nikt bezpośrednio przed tym zdarzeniem nie wychodzi z pracy.

- Bezpośrednio przed zdarzeniem „w pracy jest i informatyków” do pracy może albo przyjść pewna liczba informatyków, albo może z niej wyjść pewna liczba informatyków, ale nie może być tak, że jednocześnie pewien informatyk przychodzi, a inny wychodzi.

Uzasadnienie jest podobne jak poprzednio – każde zdarzenie nieuwzględnione powyżej możemy przesunąć do przodu w czasie. Przykładowo, jeśli jakiś informatyk inny niż X przychodzi do pracy lub wychodzi z niej bezpośrednio przed zdarzeniem „informatyk X zeznaje po raz pierwszy”, to poprawne rozwiązanie otrzymamy również, jeśli jego przyjście bądź wyjście przesuniemy bezpośrednio za to zdarzenie. Podobnie możemy przesunąć na później każdą zmianę warty następującą bezpośrednio przed liczeniem informatyków.

Kto wychodzi, kto przychodzi?

Wiemy zatem, kiedy informatycy przychodzą do pracy (albo bezpośrednio przed swoim pierwszym zeznaniem, albo przed liczeniem) i kiedy wychodzą (bezpośrednio przed liczeniem). W przypadku liczenia nie wiemy jednak, którzy informatycy przychodzą i wychodzą.

Wychodzenie jest prostsze. W momencie, w którym ktoś musi wyjść z pracy, żeby zmniejszyć liczbę obecnych informatyków, nie może to być nikt, kto jeszcze musi złożyć zeznanie. Wszyscy pozostali informatycy są z naszego punktu widzenia nierozróżnialni. Dokładniej: jeśli mamy poprawne rozwiązanie i przed pewnym liczeniem mamy w pracy x informatyków, którzy już nie będą składać zeznań, to otrzymamy również poprawne rozwiązanie, jeśli dowolnie przemieszamy godziny wyjścia tych x informatyków – poprawność gwarantuje to, że liczebności w żadnym momencie się nie zmienią, a informacji o zeznaniach składanych przez tych informatyków w przyszłości już nie ma.

Kluczem do rozwiązania zadania będzie spostrzeżenie, że możemy założyć, że informatycy przychodzą do pracy w tej samej kolejności, w której składają pierwsze zeznanie. Załóżmy, że mamy rozwiązanie R_1, w którym informatyk A składa pierwsze zeznanie przed pierwszym zeznaniem informatyka B, ale przychodzi do pracy po B. Wykażemy, że zamieniając godziny przyjścia do pracy informatyków A i B, dostaniemy rozwiązanie R_2, które również jest poprawne.

Po pierwsze, zauważmy, że godzina przybycia A do pracy w R_1 jest nie późniejsza niż pierwsze zeznanie A, które jest przed pierwszym zeznaniem B, zaś godzina przybycia B do pracy w R_1 jest jeszcze wcześniejsza niż godzina przyjścia do pracy A. Zatem w R_2 wszyscy informatycy są obecni w pracy w momentach, w których zeznają. W szczególności, każdy z informatyków przychodzi w R_2 do pracy wcześniej, niż z niej wychodzi (tutaj wykorzystujemy fakt, że B również złożył zeznanie; gdyby B nie złożył żadnego zeznania, to mogłoby się zdarzyć, że nasza zamiana byłaby niepoprawna). Po drugie, w każdej chwili liczba informatyków w biurze jest taka sama w obu rozwiązaniach, a zatem skoro R_1 było poprawne, to i R_2 jest poprawne.

Zatem, pomijając informatyków, którzy nie złożyli żadnych zeznań, możemy założyć, że informatycy przychodzą do biura w kolejności zgodnej z kolejnością składania pierwszego zeznania.

Pozostała kwestia informatyków, którzy w ogóle nie składali zeznań. Wiemy, że mogą oni wychodzić z pracy w dowolnym momencie. Pytanie, na które jeszcze nie znamy odpowiedzi, to – gdy wiemy, że jakiś informatyk musi przyjść do pracy – czy wybrać tego, który zeznanie złożył najwcześniej, czy jednego z tych, którzy w ogóle nie złożyli zeznania. Jak się jednak wkrótce okaże – nie będziemy musieli na to pytanie odpowiadać.

Rozwiązanie wzorcowe

W rozwiązaniu wzorcowym będziemy rozpatrywali chronologicznie trzy typy zdarzeń: „informatyk X zeznaje po raz pierwszy", „informatyk X zeznaje po raz ostatni" oraz „w pracy jest i informatyków". Podobnie jak pierwsze zeznanie danego informatyka rozpatrywaliśmy przed zdarzeniem „w pracy jest i informatyków" związanym z tym zeznaniem, to jego ostatnie zeznanie będziemy rozpatrywali po tym zdarzeniu.

Podzielimy naszych informatyków na pięć grup:

- *Śpiący* – informatycy, którzy na pewno jeszcze nie pojawili się w pracy.
- *Zeznający* – informatycy, którzy na pewno są w pracy, gdyż jesteśmy pomiędzy ich pierwszym a ostatnim zeznaniem.
- *Przed* – informatycy, którzy są w pracy, ale nie zeznawali (dlatego, że ich pierwsze zeznanie jeszcze nie nastąpiło albo że w ogóle nie zeznają).
- *Po* – informatycy, którzy jeszcze są w pracy, ale już złożyli ostatnie zeznanie (ta grupa nie zawiera informatyków, którzy w ogóle nie składają zeznań).
- *Wyszli* – informatycy, którzy już opuścili pracę.

W każdym momencie każdy z informatyków należy do dokładnie jednej powyższych pięciu grup. W rozwiązaniu wzorcowym nie będziemy śledzić, który z informatyków należy do której grupy, ale w każdej chwili będziemy znali *liczność* każdej z grup.

Spójrzmy, co powyższe rozważania mówią nam o przejściach informatyków między grupami w chwilach poszczególnych zdarzeń:

- W momencie zliczania informatyków, jeśli obecnie jest ich w pracy za mało, to ktoś będzie musiał przyjść do pracy – a zatem pewna liczba informatyków przejdzie z grupy *Śpiący* do grupy *Przed* (tak więc zmniejszamy liczność grupy *Śpiący* i zwiększamy liczność grupy *Przed*).
- Jeśli informatyków jest w momencie zliczania za dużo, to ktoś musi z pracy wyjść. Kandydatami są informatycy z grupy *Po* oraz pewna (nieznana nam) liczba informatyków, którzy znaleźli się w grupie *Przed*, bo nie składają żadnych zeznań. Wiemy, że poprawność rozwiązania *nie* zależy od tego, których z nich wybierzemy, więc zacznijmy od informatyków z grupy *Po*, bo wiemy, że każdy w tej grupie jest kandydatem do wyjścia z pracy. Jeśli ta grupa jest już pusta, to (o ile w ogóle istnieje poprawne rozwiązanie) w grupie *Przed* muszą znajdować się informatycy, którzy w ogóle nie składają zeznań. Możemy zatem usunąć ich z grupy *Przed*, zmniejszając liczność tej grupy.

- W momencie zdarzenia „informatyk X zeznaje po raz pierwszy” mamy dwie możliwości – albo informatyk, o którego chodzi, już był w pracy (wtedy przesuwamy go z grupy *Przed* do grupy *Zeznający*), albo nie (wtedy przesuwamy go z grupy *Śpiący* do grupy *Zeznający*). Za chwilę uzasadnimy, że jeśli tylko grupa *Przed* jest niepusta, to powinniśmy zmniejszyć liczność tej właśnie grupy.

- Najprostsze jest radzenie sobie ze zdarzeniem „informatyk X zeznaje po raz ostatni” – wystarczy przełożyć jednego informatyka z grupy *Zeznający* do grupy *Po*.

Przy założeniu, że udowodnimy ostatni brakujący fakt, algorytm rozwiązania zadania jest już jasny. Dla każdego informatyka zapamiętujemy jego najwcześniejsze i najpóźniejsze zeznanie, a następnie tworzymy zbiór zdarzeń i przechodzimy przez te zdarzenia chronologicznie. Początkowo grupa *Śpiący* zawiera n informatyków, a pozostałe grupy są puste. W każdym momencie pamiętamy liczność każdej z pięciu grup i poprawiamy ją według powyższych zasad.

Mogą wystąpić trzy przypadki, które oznaczają, że rozwiązanie nie istnieje:

- Jeśli w momencie zliczania informatyków chcemy dodać informatyka, a grupa *Śpiący* jest już pusta, lub

- jeśli w momencie pierwszego zeznania chcemy dodać informatyka, a zarówno grupa *Przed* jak i grupa *Śpiący* są puste, lub

- jeśli w momencie zliczania chcemy usunąć informatyka, a zarówno grupa *Przed* jak i grupa *Po* są puste.

Jeśli nasz algorytm zakończy się sukcesem, to łatwo będzie nam wskazać faktycznie poprawne rozwiązanie. Trudniejsze jest uzasadnienie, że jeśli istnieje poprawne rozwiązanie, to nasz algorytm je znajdzie. Kluczowe tu jest sprawdzenie, że możemy założyć, że jeśli przekładamy jakiegoś informatyka do grupy *Zeznający*, a grupa *Przed* jest niepusta, to możemy go przełożyć z grupy *Przed*. Zauważmy jednak, że jeśli mamy rozwiązanie, a w nim informatyka B, który nie składa żadnych zeznań, przychodzi do pracy przed informatykiem A, który składa zeznania, i B jest wciąż obecny w momencie, gdy A przychodzi do pracy, to możemy zamienić godziny przyjścia do pracy informatyków A i B i wciąż mieć poprawne rozwiązanie (a następnie przesunąć moment przyjścia B do pracy jeszcze później, po zdarzeniu „informatyk A zeznaje po raz pierwszy”). Możemy zatem faktycznie założyć, że jeśli ktokolwiek jest w grupie *Przed*, to jest tam między innymi informatyk, który zaczął teraz zeznawać. Zatem faktycznie, jeśli tylko istnieje poprawne rozwiązanie, to nasz algorytm je znajdzie.

Łączny czas działania wzorcowego algorytmu to $O((n+m)\log m)$ – w każdym z $O(\log m)$ kroków wyszukiwania binarnego będziemy musieli obliczyć w czasie $O(n+m)$ godziny pierwszego i ostatniego zeznania każdego informatyka, następnie w czasie $O(m)$ posortować kubełkowo wszystkie $O(m)$ zdarzeń w porządku chronologicznym i w czasie $O(m)$ przejść przez nie, aktualizując liczności grup informatyków.

Rozwiązanie wzorcowe zaimplementowano w plikach `ins.cpp` oraz `ins1.pas`.

Testy

Przygotowanie testów okazało się zadaniem nietrywialnym. W przypadku losowych testów przeróbki rozwiązania wzorcowego zawierające pojedynczy błąd przechodziły zdecydowaną większość testów.

Z tego powodu konieczne było umieszczenie wielu przypadków testowych w jednym pliku oraz bardziej przemyślany sposób generowania testów. Dużej liczby przypadków testowych nie dało się uniknąć, gdyż każdy test sprawdza co najwyżej jedną ścieżkę w rozwiązaniu – tę odpowiedzialną za pierwszą napotkaną sprzeczność. Stworzono 13 testów składających się z 50 przypadków testowych każdy.

Testy generowane były w sposób losowy z zapewnieniem następujących warunków: żądana liczba informatyków, żądana liczba zeznań, żądana liczba chwil, w których zostało złożone przynajmniej jedno zeznanie, oraz ograniczenie dolne na wynik.

Dodatkowo zapewniono, że w niektórych testach bardzo wiele zeznań złożonych jest w jednej chwili oraz że istnieje informatyk, który złożył bardzo wiele zeznań. Ponadto w większości testów wszystkie zeznania zgadzają się co do liczby osób, które były w biurze w poszczególnych chwilach. W części testów dodano informatyków, którzy nie składają żadnego zeznania.