Algorytmy i struktury danych, pracownia specjalistyczna, studia stacjonarne, zestaw zadań nr 1
Krzysztof Ostrowski

Ćwiczenia w optymalizacji rozwiązań algorytmicznych pod względem złożoności czasowej. Należy:

1) **Zaimplementować i przetestować dwa różne rozwiązania dla jednego z problemów, jedno z nich powinno mieć możliwie najniższą złożoność czasową (oraz czas wykonania)**
2) **Przygotować sprawozdanie zawierające opis programów (algorytmów), obliczenie ich złożoności czasowej oraz porównanie ich czasów wykonania dla rosnących rozmiarów danych (tabele, wykres) oraz wnioski.**
3) **W rozwiązaniach tego zestawu nie można korzystać z gotowych kontenerów i algorytmów**

Problem 1 – Analiza obrazu

Pewna firma informatyczna zajmuje się analizą obrazów medycznych. Aktualnie potrzebne jest specjalistyczne oprogramowanie pozwalające wykrywać pewne cechy obrazu. Po wstępnej obróbce piksele obrazu zostały podzielone na jaśniejsze (0) i ciemniejsze (1). Celem jest wykrywanie dużych kwadratowych i ciemnych obszarów. Jesteś pracownikiem tej firmy i twoim zadaniem jest napisanie wydajnego algorytmu rozwiązującego ten problem.

Wejście:
W pierwszej linii wejścia podany jest rozmiar *n (*1<=*n*<=3000) kwadratowego obrazu. W kolejnych *n* liniach wejścia znajdują się znaki (po *n* w każdej linii) oznaczające oznaczające jasny (0) lub ciemny (1) piksel. W ostatniej linii znajduje się liczba całkowita *m* (1<=*m*<=*n*) oznaczająca rozmiar analizowanego kwadratowego fragmentu obrazu.

Wyjście:
W pierwszej linii wyjścia ma pojawić się liczba całkowita oznaczająca największą liczbę ciemnych pikseli jaką posiada pewien fragment obrazu (o zadanym rozmiarze).

Przykład:
Wejście (in.txt):
5 //rozmiar obrazu (5x5)
10010
11101
10100
01111
11001
3 //analizowany fragment rozmiaru 3x3

Wyjście (out.txt):
7 //tyle ciemnych pikseli ma fragment o największej ich liczbie (pogrubiony)

Problem 2 - Kody

Pan Gniewomir jest specjalistą od kodowania, ale ma awersję do komputerów, co jest zapewne spowodowane jego bardzo zaawansowanym wiekiem. Jego papierowa baza danych zawierająca różne kody rozszerza się jednak coraz bardziej i przestaje on nad tym panować. Nie starcza mu

czasu, by wszystko ogarnąć, a i on chce czasem odpocząć i się rozerwać w klubie dla seniorów. Stąd potrzebna mu pomoc informatyka. Zaimplementuj algorytm, który będzie obliczał częstość występowania kodów binarnych. Długość kodu może wynosić maksymalnie 25 bitów, a każdy kod zaczyna się jedynką.

Wejście:
W pierwszym wierszu wejścia podana jest liczba *n* ($1<=n<=10^6$) oznaczająca liczność zbioru kodów. W kolejnych *n* wierszach wejścia podanych jest *n* kodów.

Wyjście:
W jedynej linii wyjścia ma się pojawić liczba wystąpień najczęściej występującego kodu w zbiorze.

Przykład:
Wejście (in.txt):
7 //7 kodów w zbiorze
10010 //kody ze zbioru
101
11100
1001
10010
101
10010

Wyjście (out.txt):
3 //jeden z kodów (10010) powtarza się aż 3 razy

Problem 3 – Misja

Ziemia weszła w konflikt z obcą cywilizacją, która posiada wyjątkowo niebezpieczną broń mogącą zniszczyć naszą planetę. Kapitan kosmicznego statku „Enterprise” ma za zadanie unieszkodliwić wroga. Posiada on rozmieszczenie baz nieprzyjaciela i planuje im przeszkodzić. W tym celu zostanie użyta nowatorska broń nadprzestrzenna, która niszczy wszystkie obiekty położone w zasięgu jej wiązki (jej szerokość kątowa to 90 stopni). Mając do dyspozycji współrzędne kątowe baz nieprzyjaciela należy wyznaczyć optymalny punkt ulokowania broni, by unieszkodliwić jak najwięcej baz za jednym wystrzałem. Dzięki swoim nadzwyczajnym umiejętnościom algorytmicznym możesz uratować ziemię!

Wejście:
W jedynej linii wejścia podana jest liczba *n* baz nieprzyjaciela ($1<=n<=10^6$). W kolejnych *n* liniach podane są po dwie liczby całkowite oznaczające współrzędne kątowe kolejnych baz (stopnie od 0 do 359, minuty od 0 do 59) w stosunku do miejsca przetrzymywania broni (mogą się one powtarzać). Możemy założyć, że kąt 0 stopni to północ, 45 stopni to północny-wschód, 90 stopni to wschód itd.

Wyjście:
W pierwszym wierszu wyjścia podana jest maksymalna możliwa liczba zniszczonych baz, jeśli broń będzie ustawiona pod odpowiednim kątem

Przykład:
Wejście (in.txt):
7 //7 baz
170 0 //pierwsza baza ma wsp. kątową 170 stopni i 0 minut
95 15 //itd.
0 5
260 0
70 23
190 0
330 38

Wyjście (out.txt):
3 //przy odpowiednim ustawieniu broni można zniszczyć bazy o wsp. 170, 190 i 260 st

Problem 4 – Rejestr mieszkańców

W pewnym kraju niektórzy bogacili się na potęgę, a inni z kolei ledwo wiązali koniec z końcem. W końcu władze postanowiły coś z tym zrobić – bogaci mieli płacić biednym. Wpierw jednak każdy mieszkaniec państwa był zobowiązany ujawnić swój majątek. Następnie zatrudniono informatyków i statystyków by przeanalizowali dane i wprowadzili odpowiedni podatek. W tym celu potrzebne było obliczenie liczby mieszkańców należących do odpowiednich widełek majątkowych (posiadających od x do y pieniędzy). Niestety sprawa nie jest prosta, gdyż danych jest bardzo dużo. Zostałeś zatrudniony jako informatyk i masz opracować wydajny program wspomagający analizę danych. Jeżeli nie zawiedziesz, to władze się hojnie odwdzięczą (a może nawet zaliczysz ASD).

Wejście:

W pierwszy wejściu podana jest liczba *n* mieszkańców kraju ($1<n<=10^6$) oraz liczba zapytań *m* (1<=*m*<=n). W kolejnych *n* liniach podane są majątki kolejnych mieszkańców kraju (majątek to liczba całkowita nie większa od 10^{18}). W kolejnych *m* liniach podane są zapytania: po dwie liczby całkowite oznaczające interesujące nas widełki majątkowe (pierwsza liczba mniejsza lub równa drugiej liczbie).

Wyjście:

W kolejnych m liniach wyjścia podane są liczby mieszkańców mieszczących się w poszczególnych widełkach majątkowych

Przykład:

Wejście (in.txt):
10 3 //liczba mieszkańców i zapytań
23 //majątek pierwszego mieszkańca
14 //itd
4
81
75
102
35
7
1005

1
99 1007 //pierwsze widełki majątkowe
3 19 //itd
500 700

Wyjście (out.txt):
2
3
0

Problem 5 – Matematyk

Jaś i jego koledzy mają problem z matematyką. Zrzędliwy nauczyciel zadaje im coraz trudniejsze zadania. Najtrudniejsze z nich są związane z obliczaniem sumy przedziałów liczbowych postaci <a, b>. Dziadek Jasia od miesiąca siedzi i oblicza odpowiedzi. W trosce o zdrowie dziadka Jasio postanowił skorzystać z pomocy komputera, ale to nie takie proste! Na szczęście zwrócił się do swojego kuzyna programisty, czyli do Ciebie. Pomóż Jasiowi i jego kolegom przy okazji ratując życie staruszka.

Wejście:

W pierwszej linii podana jest liczba *n* ($n<=10^6$) przedziałów. W kolejnych n liniach podane są po dwie liczby całkowite (z zakresu od 0 do 10^{18}). oznaczające końce kolejnych przedziałów (zakładamy, że koniec przedziału jest zawsze większy niż początek).

Wyjście:

W kolejnych liniach wyjścia powinny znaleźć się przedziały (kolejno ułożone) wchodzące w skład obliczanej sumy.

Przykład:

Wejście (in.txt)
5
1 3
5 9
2 4
11 13
8 10

Wyjście (out.txt)
1 4
5 10
11 13