Zajęcia 29

Temat: Algorytmy tekstowe
Czas trwania: 2x45 min
Cel zajęć:

projektuje i programuje proste problemy z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, tablice, rekurencję, pisze własne funkcje rekurencyjne, struktury danych, biblioteka STL, algorytmy tekstowe, łańcuchy znaków, testuje poprawność programów dla różnych danych, posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;

Efekty:

- umie uruchomić potrzebne oprogramowanie,
- umie napisać program z wykorzystaniem algorytmów tekstowych,
- potrafi wyszukiwać wzorzec w tekście,
- zna algorytm naiwny i KMP,

Formy i metody pracy: praca samodzielna, omówienie, wykład

Zadania do wykonania na zajęciach	Treści programowe
1. Czy tu już byłem?	M.3, P.2.15, A.3.3,
2. Numer telefonu	M.3, P.2.15, A.3.3,

Materiały do zajęć:
http://informatykaplus.edu.pl/upload/list/czytelnia/Przeglad_podstawowych_algorytmow.pdf s. 28

Zadania do wykonania w domu:

Szablon

https://szkopul.edu.pl/problemset/problem/a3IarwgOdubufXQ89OsQz3v_/site/

ZADANIA I ROZWIĄZANIA

Zadanie 1. Numer telefonu

Dostępna pamięć: 256MB

W Bajtolandii każdy numer telefonu składa się tylko z cyfr 3 oraz 5 i zawsze zaczyna się od 3. Bitek nie lubi pamiętać długich i monotonnych numerów - zamiast tego woli zapamiętywać prefikso-sufiksy dla każdej z cyfr. Zastanawia się teraz, jak odtworzyć numer telefonu i czy dobrze zapamiętał numer.

Wejście

W pierwszej linii wejścia znajduje się jedna liczba całkowita n ($1 \le n \le 10^6$), oznaczająca długość tablicy prefikso-sufiksów. W drugiej linii wejścia znajduje się n-elementowa tablica prefikso-sufiksów s_i ($0 \le s_i \le 10^6$), a jej elementy są oddzielone pojedynczymi odstępami.

Wyjście

Na wyjściu powinien w pierwszej linii znaleźć się n-cyfrowy numer telefonu odtworzony z tablicy prefikso-sufiksów lub napis NIE - odpowiedź na pytanie, czy tablica była poprawnie zapamiętana.

Przykład

Wejście	Wyjście
8 0 0 1 1 2 3 2 3	35335353

Wejście	Wyjście
3 0 0 2	NIE

Rozwiązanie

Zacznijmy od konstrukcji numeru telefonu. Pierwszą jego cyfrą jest 3. Kolejne cyfry zależą od tablicy prefikso-sufiksów. Jeżeli wartość w tablicy na i-tej pozycji wynosi 0, oznacza to, że nie mamy żadnej cyfry zgodnej z początkiem, więc musimy użyć cyfry 5. Dla wartości `pref[i]` z tablicy prefikso-sufiksów do numeru telefonu dodamy znak o indeksie `pref[i]` z dotychczasowego numeru telefonu.

```
wynik[] // tablica, w której zapamiętamy numer telefonu
wynik[0] ← 3
dla i=1,2,…,n-1 wykonuj
    jeżeli (pref[i] = 0)
        wynik[i] ← 5
    przeciwnie
        wynik[i] ← pref[wynik[i]]
```

Teraz obliczmy tablicę prefikso-sufiksów `K[]` (algorytm KMP) dla obliczonej tablicy `wynik[]`.

```
K[0] ← 0
dla i=1,1,2,...,n-1 wykonuj
    //zakładamy zgodność dla następnej cyfry
    K[i] ← K[i-1]
    // jeżeli litery w naszym zapisie się nie zgadzają,
    //zmniejszamy wartość aktualnego na poprzedni  prefikso-sufiks
    dopóki (K[i]>0 ∧ wynik[K[i]] ≠ wynik[i])
        K[i] ← K[K[i]-1]
    //jeżeli bieżąca litera się zgadza,
    //zwiększamy długość aktualnego prefikso-sufiksu o 1
    jeżeli (wynik[K[i]] = wynik[i])
        K[i] ← K[i]+1
```

Teraz wystarczy sprawdzić, czy tablica `K[]` oraz `pref[i]` są równe.

Zadanie 2. Czy tu już byłem?

Dostępna pamięć: 256MB

Bajtek zwiedzał Bajtogród przez cały dzień. Wyraźnie widać już po nim zmęczenie. Podobają się mu jednak tutejsze skrzyżowania. Wokół każdego z nich stoi zawsze dokładnie n budynków. Skrzyżowania różnią się od siebie tylko wysokościami poszczególnych domów. Każde skrzyżowanie jest otoczone budynkami o charakterystycznym dla siebie ciągu wysokości.

Bajtek znalazł się właśnie na kolejnym skrzyżowaniu, ale w żaden sposób nie może sobie przypomnieć, czy wcześniej już tu był. Na szczęcie każde skrzyżowanie, które dzisiaj odwiedził, opisał w zeszycie. Który to już raz Bajtek odwiedził aktualne skrzyżowanie?

Wejście

W pierwszej linii wejścia znajdują się dwie liczby całkowite n oraz k ($1 \le n \le 10^5$,$1 \le k \le 100$), oznaczające odpowiednio liczbę budynków na każdym ze skrzyżowań oraz liczbę skrzyżowań, które wcześniej odwiedził Bajtek. W drugiej linii wejścia znajduje się n-elementowy ciąg wysokości kolejnych budynków na aktualnym skrzyżowaniu s_i ($1 \le s_i \le 10^6$), a jego elementy są oddzielone pojedynczymi odstępami. W kolejnych k liniach znajdują się opisy odwiedzonych wcześniej przez Bajtka skrzyżowań. Każdy opis to n-elementowy ciąg wysokości kolejnych budynków na jednym z odwiedzonych skrzyżowań o_i ($1 \le o_i \le 10^6$).

Wyjście

Na wyjściu w kolejnych k liniach wypisz komunikat TAK lub NIE - odpowiedź na pytanie, czy Bajtek był już wcześniej na tym skrzyżowaniu. Bajtek mógł odwiedzić aktualne skrzyżowanie wcześniej więcej niż raz.

Przykład

Wejście	Wyjście
5 3 1 2 3 4 5 2 3 4 5 1 3 4 5 1 2 3 5 4 1 2	TAK TAK NIE

Ocenianie

Podzadanie	Ograniczenia	Liczba punktów
1	$n, k \le 10^2$	25
2	brak dodatkowych założeń	75

Rozwiązanie

W zadaniu musimy sprawdzić równoważność cykliczną dwóch słów. Najprostszym sposobem będzie wyszukanie wzorca `w` w tablicy `tt` (podwójna tablica `t`). Możemy tu użyć algorytmu KMP na tablicach `w+t+t` (+ oznacza połączenie tablic w jedną).

```
wynik[] //tablica złożona z tablic w, t oraz t;
```

```
        //pomiędzy tablicę w i t dopiszemy wartość -1
K[0] ← 0
dla i=1,2,…,3·n wykonuj
    //zakładamy zgodność dla następnej cyfry
    K[i] ← K[i-1]
    // jeżeli litery w naszym zapisie się nie zgadzają,
    //zmniejszamy wartość aktualnego na poprzedni prefikso-sufiks
    dopóki (K[i]>0 ∧ wynik[K[i]] ≠ wynik[i])
        K[i] ← K[K[i]-1]
    //jeżeli bieżąca litera się zgadza,
    //zwiększamy długość aktualnego prefikso-sufiksu o 1
    jeżeli (wynik[K[i]] = wynik[i])
        K[i] ← K[i]+1
```

Jeżeli teraz w tablicy prefikso-sufiksów `K[]` na pozycjach większych niż `|w|` znajdziemy prefikso-sufiks o długości `|w|`, oznacza to, że znaleźliśmy się na już odwiedzonym wcześniej skrzyżowaniu.

Można również użyć algorytmu działającego także w czasie O(n) i nie zużywającego dodatkowej pamięci. Jego opis można znaleźć tu:

http://www.rafalnowak.pl/wiki/index.php?title=R%C3%B3wnowa%C5%BCno%C5%9B%C4%87_cykliczna_dw%C3%B3ch_s%C5%82%C3%B3w