

Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Zadanie Wspinaczka – LOGIA 22 (2021/22), etap 3

Treść zadania

Grześ chce wyznaczyć najlepszą trasę na ścianie wspinaczkowej w kształcie walca. Posiada opis ścianki w postaci prostokątnej planszy. Naniesione są na niej punkty, które się zdobywa przechodząc przez dane pole. Grześ rozpoczyna wspinaczkę od dowolnego pola w najniższym rzędzie. W każdym kroku wybiera w rzędzie powyżej jedno z trzech pól sąsiadujących z polem, na którym przebywa. Skrajne lewe pole pozwala na przeniesienie się na skrajne prawe pole w rzędzie powyżej i odwrotnie, skrajne prawe pozwala przenieść się na skrajne lewe.

Pomóż Grzysiowi i napisz program liczący maksymalną liczbę punktów, które można uzyskać.

9	3	3	4	7	8
6	7	2	5	6	8
4	1	4	7	7	8
1	6	5	1	7	9
6	6	7	5	7	1
5	7	9	8	3	7

Dla planszy obok najlepsza droga to:
 $8 + 7 + 9 + 8 + 8 + 9 = 49$.

Wejście:

Pierwszy wiersz wejścia zawiera dwie liczby naturalne: **w** – liczbę rzędów i **k** – liczbę kolumn, oddzielone spacją, $2 \leq w, k \leq 100$

Każdy z kolejnych **w** wierszy wejścia zawiera **k** liczb naturalnych z przedziału $[1; 100]$ rozdzielonych spacją, opisujących kolejne rzędy planszy poczynając od górnego rzędu.

Wyjście:

Liczba naturalna – maksymalna liczba punktów do uzyskania.

	Przykład 1	Przykład 2	Przykład 3
Wejście	6 6 9 3 3 4 7 8 6 7 2 5 6 8 4 1 4 7 7 8 1 6 5 1 7 9 6 6 7 5 7 1 5 7 9 8 3 7	4 8 31 49 96 20 88 96 12 46 30 58 95 72 55 29 44 90 23 8 56 96 3 91 54 77 85 45 4 80 20 52 58 95	8 5 55 49 43 25 83 46 28 18 10 10 16 96 62 12 80 40 81 92 45 5 47 35 31 21 28 23 72 36 3 71 81 53 76 63 91 93 94 74 100 46
Wyjście	49	367	615
Komentarz	$8+7+9+8+8+9$	$80 + 96 + 95 + 96$	$100 + 91 + 71 + 47 + 81 + 96 + 46 + 83$



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Omówienie rozwiązania

Rozwiązanie zadania poprzedzimy kilkoma obserwacjami:

1. wynik nie ulega zmianie, gdy wiersze (rzędy) będą w odwróconej kolejności,
2. gdy mamy tylko jeden wiersz danych, to wynikiem jest największa wartość w tym wierszu,
3. dwa rzędy można „zredukować” do jednego z wartościami będącymi sumą elementu górnego oraz maksymalnego z trzech elementów sąsiednich poniżej, na przykład:

3	5	7	9	11	13
3	4	5	6	7	8

zastąpimy

11	10	13	16	19	21
----	----	----	----	----	----

należy pamiętać, że sąsiednimi są też skrajne elementy,

4. mając wiele rzędów, można zmniejszyć ich liczbę o jeden zastępując dwa dolne rzędy przez jeden zgodnie z obserwacją 3, a wynik pozostanie ten sam.

Mając poczynione powyższe obserwacje widzimy, że wyniku można zacząć szukać bez potrzeby wczytania wszystkich danych. Wczytując drugi wiersz obliczamy jednocześnie wartości do wiersza, którym zastąpimy poprzedni i wczytywany.

Rozwiązanie w postaci listy kroków:

```
wczytaj w, k
zainicjuj tablicę punktyWspinaczki[0..k-1] zerami
zainicjuj tablicę tmp[0..k-1] zerami
dla j z [0, 1, 2, ..., w - 1] wykonuj kroki 5-8
    dla i z [0, 1, 2, ..., k - 1] wykonuj kroki 6-7
        wczytaj d
        tmp[i] = d + maksimum(punktyWspinaczki[(i-1+k) mod k],
                               punktyWspinaczki[i],
                               punktyWspinaczki[(i+1) mod k])
    punktyWspinaczki = tmp
wyświetl maksimum(punktyWspinaczki)
```

W pierwszych trzech liniach wczytujemy liczbę rzędów i kolumn w, k oraz ustawiamy początkowe wartości tablic punktyWspinaczki i tmp na zero. W liniach 4-8 wczytywany są dane i obliczany wiersz do tablicy pomocniczej tmp, który zastąpi dwa wiersze. Nie można aktualizować od razu wartości w tablicy punktyWspinaczki, bo nadpisywana wartość jest potrzebna do obliczeń przy następnej iteracji pętli wewnętrznej. W linii 7 użyto wyrażenia $i-1+k$, by dzielna była zawsze dodatnia. Linia 8 to przepisanie wartości do tablicy punktyWspinaczki, która będzie teraz zawierać już wiersz zastępujący dwa poprzednie.



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Rozwiązanie w języku Python

Wykorzystując rozwiązanie w postaci listy kroków gotowy program mógłby wyglądać tak:

```
1 w, k = map(int, input().split())
2 punktyWspinaczki = [0] * k
3 tmp = [0] * k
4 for j in range(w):
5     wiersz = input().split()
6     for i in range(k):
7         d = int(wiersz[i])
8         tmp[i] = d + max(punktyWspinaczki[i - 1],
9                           punktyWspinaczki[i],
10                          punktyWspinaczki[(i + 1) % k])
11     punktyWspinaczki = tmp[:]
12 print(max(punktyWspinaczki))
```

W linii 1 wczytane dane tekstowe z wiersza są rozdzielane na listę a następnie funkcja `map()` dla każdego elementu z tej listy wywołuje funkcję `int()`. Pierwszy wiersz danych zawiera dwie wartości opisujące dane i przypisujemy je do zmiennych `w` i `k`. Linie 2 i 3 inicjalizują zerami `k` elementowe listy `punktyWspinaczki` i `tmp`. Linia 5, to wczytanie całego wiersza danych jako napisu i rozdzielenie na elementy pamiętane w liście `wiersz`, by w linii 7 móc pobierać `i`-tą wartość z wiersza. Wyrażenie $(i-1+k) \bmod k$ w linii 8 możemy zastąpić wyrażeniem `i-1`, gdyż dla `i=0` otrzymujemy wartość wskazującą na ostatni element tablicy. W linii 9 przepisywana jest lista `tmp` do `punktyWspinaczki`. Ważne jest by użyć znaków `[:]`, lub metody `tmp.copy()` – w przeciwnym przypadku nazwa `punktyWspinaczki` będzie wskazywać na listę `tmp` i zmiana wartości `tmp` będzie skutkowałą zmianą wartości `punktyWspinaczki`. Ostatnia linijka, to znalezienie wartości największej z listy `punktyWspinaczki` i jej wyświetlenie.

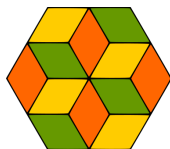
Poniżej przytaczamy jeszcze rozwiązanie bez wykorzystania pomocniczej tablicy:

```
1 w, k = map(int, input().split())
2 punktyWspinaczki = [0]*k
3 for _ in range(w):
4     punktyWspinaczki = [max(punktyWspinaczki[i-1],punktyWspinaczki[i],
5                             punktyWspinaczki[(i+1)%k]) + int(d) for i, d in enumerate(input().split())]
6     print(max(punktyWspinaczki))
```

Korzystając z listy składanej nie potrzebujemy pomocniczej tablicy `tmp`. Funkcja `enumerate()` generuje pary wartości, w których pierwsza wartość jest numerem porządkowym z domyślnym początkiem w zerze, a druga wartość jest elementem listy. Numer porządkowy jest przypisany do zmiennej `i`, a w zmiennej `d` zapisywane są odczytane wartości z listy utworzonej przez funkcję `input().split()`.

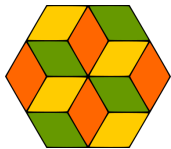
Testy

W pierwszych trzech grupach testów użyto niewielkich danych testowych, a ponadto w grupach I i II nie było w danych przechodzenia między kolumną 0 i `k-1`. Testy w grupach IV i V są losowe i na dużych danych, dlatego zawarte są w oddzielnych plikach.



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Grupa testów	Test	Wynik
I	2 2 86 89 100 12	189
	4 4 35 46 12 23 32 77 11 41 93 44 20 31 23 68 63 1	284
	5 5 32 90 22 65 4 51 19 25 96 97 68 22 29 94 95 53 54 62 53 60 70 1 33 91 34	409
II	3 5 12 58 12 73 38 50 61 51 27 49 18 18 38 91 16	215
	2 7 12 90 62 59 4 67 28 48 41 19 19 17 53 29	138
	6 16 89 36 67 41 71 58 32 29 80 62 15 14 6 30 75 37 65 20 49 30 61 65 80 3 18 11 24 72 58 26 41 81 67 20 7 9 91 98 12 65 16 55 73 20 83 51 73 94 66 28 58 73 73 87 9 59 24 18 86 80 80 55 49 84 23 83 5 19 39 30 87 37 16 3 30 62 41 91 1 76 26 25 89 20 43 99 66 93 22 48 27 25 52 79 15 84	509
III	5 5 60 23 75 14 43 51 51 2 6 70 62 9 14 20 86 1 63 64 85 68 55 69 7 39 48	349
	10 10 83 48 77 81 95 63 29 78 31 95 20 69 25 11 39 79 64 59 15 22 100 69 50 66 46 47 3 93 39 36 35 31 63 72 80 5 17 98 1 25 83 16 43 6 69 38 40 78 91 26 85 30 86 17 23 56 98 79 97 66 92 27 79 25 34 30 91 32 66 21 80 29 29 31 3 28 17 73 29 42 94 72 90 51 51 99 42 36 59 18 46 44 45 53 61 45 87 46 19 64	820
	15 15 91 84 81 36 23 1 40 22 59 38 88 32 8 61 49	1221



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

	67 32 55 22 39 40 47 60 80 45 14 35 29 8 51 40 38 69 92 69 98 87 71 7 40 16 95 70 29 53 25 99 30 98 94 30 20 40 52 71 14 21 3 15 26 95 40 42 75 82 53 2 38 30 82 2 14 14 40 37 77 64 1 53 80 86 79 33 25 49 30 27 61 71 89 65 42 94 73 20 10 53 63 61 48 59 57 65 52 38 46 8 51 84 86 79 76 32 19 11 80 2 57 68 92 19 49 61 81 76 3 47 35 28 84 91 79 18 72 49 66 73 83 3 12 70 56 19 92 17 87 92 65 66 98 41 45 61 61 92 11 1 46 34 57 22 58 21 46 58 48 91 44 80 26 56 13 31 8 17 92 26 78 34 94 7 74 12 73 38 75 56 3 85 59 82 94 48 84 47 69 94 45 50 81 86 80 46 1 31 67 55 31 65 87 11 52 46 66 97 43 70 53 54 49 73 73 30 69 80	
IV	Dane załączone w pliku, tablica 100 wierszy x 3 kolumny	7404
	Dane załączone w pliku, tablica 3 x 100	289
	Dane załączone w pliku, tablica 50 x 50	4208
V	Dane załączone w pliku, tablica 80 x 80	6753
	Dane załączone w pliku, tablica 90 x 90	7445
	Dane załączone w pliku, tablica 100 x 100	8219