

Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Zadanie Ogrodzenie – LOGIA 23 (2022/23), etap 2

Treść zadania

Zosia zaprojektowała ogrodzenie dookoła swojego ogrodu. Składa się ono z paneli różnej wysokości. Zapomniała jednak o dwóch furtkach, które planowała umieścić w ogrodzeniu. Zastanawia się teraz, które panele opłaca się zastąpić furtkami. Chce wymienić dwa panele na furtki, ale tak, by furtki nie sąsiadowały ze sobą oraz koszt ogrodzenia był jak najniższy. Koszt panelu jest równy jego wysokości, a furtki Zosia otrzymała w prezencie za darmo. Napisz program, który obliczy koszt ogrodzenia po wymianie paneli na furtki.

Wejście

Pierwszy wiersz wejścia zawiera jedną liczbę naturalną n oznaczającą liczbę paneli w ogrodzeniu, $5 \leq n \leq 1000$.

Drugi wiersz zawiera n liczb naturalnych z przedziału $[1; 10000]$ rozdzielonych spacją, opisujących wysokości kolejnych paneli w ogrodzeniu.

Wyjście

Liczba naturalna – minimalny koszt ogrodzenia.

Przykłady:

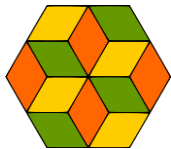
Wejście	10 5 5 6 7 3 8 5 3 5 3	5 5 5 5 7 8	5 7 5 5 5 8
Wyjście	35	17	17
Komentarz	furtki zastąpiły panele o wysokości 7 i 8 koszt ogrodzenia: $5 + 5 + 6 + 0 + 3 + 0 + 5 + 3 + 5 + 3 = 35$	furtki zastąpiły panele o wysokościach 5 i 8 uwaga: panele o wysokościach 7 i 8 sąsiadują ze sobą!	furtki zastąpiły panele o wysokościach 5 i 8 uwaga: panele o wysokościach 7 i 8 sąsiadują ze sobą!

Omówienie rozwiązania

Na pierwszy rzut oka rozwiązanie polega na znalezieniu dwóch maksymalnych wartości w podanym ciągu. Niestety jest jeden dodatkowy warunek – poszukiwane wartości nie mogą ze sobą sąsiadować. Dotyczy to również pierwszego i ostatniego elementu. Czy uwzględniając dodatkowy warunek wystarczy poszukać trzech maksymalnych wartości? Otóż nie. Poniższy przykład to demonstruje:

1 100 10000 101 2

Wyszukując trzy największe wartości otrzymamy 10000, 101 i 100. Po uwzględnieniu dodatkowego warunku wybieramy liczby 100 i 101 (nie sąsiadują ze sobą) i w wyniku otrzymujemy 10003 ($1 + 10000 + 2$). Nie jest to poprawna odpowiedź, ponieważ bardziej opłaca się usunąć z ciągu



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

10000 i 2. Wtedy wynik to 203 ($1 + 100 + 101$). Z tej prostej obserwacji wynika, że do ustalenia poprawnego wyniku poszukujemy czterech największych wartości, spośród których wybierzemy dwie niesąsiadujące ze sobą o maksymalnej sumie. Żeby prawidłowo ustalić wynik musimy oprócz liczby zapamiętywać również jej pozycję w ogrodzeniu.

Zacznijmy od najwolniejszego rozwiązania. Zbadajmy wszystkie pary liczb niesąsiadujących ze sobą w poszukiwaniu maksymalnej sumy dwóch liczb.

```
maksymalna ← 0
dla i od 0 do n - 3 wykonuj
    dla j od i + 2 do n - 1 wykonuj
        jeżeli nie (i = 0 oraz j = n - 1) oraz p[i] + p[j] > maksymalna to
            maksymalna = p[i] + p[j]
```

Rozwiązanie w języku Python

```
1 n = int(input())
2 lista = input().split()
3 p = []
4 suma = 0
5 for i in range(n):
6     p.append(int(lista[i]))
7     suma = suma + int(lista[i])
8 maksymalna = 0
9 for i in range(n - 2):
10     for j in range(i + 2, n):
11         if not (i==0 and j==n-1) and p[i] + p[j] > maksymalna:
12             maksymalna = p[i] + p[j]
13 print(suma-maksymalna)
```

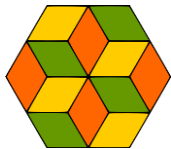
Powyższe rozwiązanie jest proste do implementacji i ma złożoność czasową wystarczającą do danych z treści zadania (1000 liczb).

Szybsze rozwiązanie sprowadza się do znalezienia czterech największych wartości.

```
1 maks4 = [[0, 0], [0, 0], [0, 0], [0, 0]]
2 for i in range(n):
3     if p[i] > maks4[0][0]:
4         maks4.insert(0, [p[i], i])
5         maks4.pop()
6     elif p[i] > maks4[1][0]:
7         maks4.insert(1, [p[i], i])
8         maks4.pop()
9     elif p[i] > maks4[2][0]:
10        maks4.insert(2, [p[i], i])
11        maks4.pop()
12    elif p[i] > maks4[3][0]:
13        maks4.insert(3, [p[i], i])
14        maks4.pop()
```

Zauważmy, że tylko raz przeglądaliśmy zbiór danych.

Po ustaleniu czterech największych wartości (lista maks4) można sprawdzić, które liczby będą stanowiły rozwiązanie. Do rozpatrzenia jest sześć par (0, 1), (0, 2), (0, 3), (1, 2), (1, 3) i (2, 3).

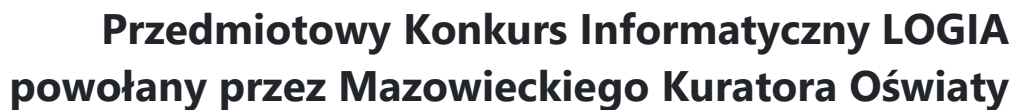


Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

```
1 maksymalna = 0
2 for i in range(3):
3     for j in range(i + 1, 4):
4         if (not (maks4[i][1]==0 and maks4[j][1]==n-1)) and
            (not (maks4[i][1]==n-1 and maks4[j][1]==0)) and
            abs(maks4[i][1] - maks4[j][1]) > 1 and
            maks4[i][0] + maks4[j][0] > maksymalna:
5             maksymalna = maks4[i][0] + maks4[j][0]
```

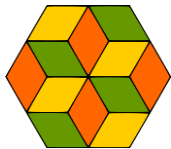
Poniżej kod całego programu.

```
1 n = int(input())
2 lista = input().split()
3 p = []
4 suma = 0
5 for i in range(n):
6     p.append(int(lista[i]))
7     suma = suma + int(lista[i])
8 maks4 = [[0, 0], [0, 0], [0, 0], [0, 0]]
9 for i in range(n):
10     if p[i] > maks4[0][0]:
11         maks4.insert(0, [p[i], i])
12         maks4.pop()
13     elif p[i] > maks4[1][0]:
14         maks4.insert(1, [p[i], i])
15         maks4.pop()
16     elif p[i] > maks4[2][0]:
17         maks4.insert(2, [p[i], i])
18         maks4.pop()
19     elif p[i] > maks4[3][0]:
20         maks4.insert(3, [p[i], i])
21         maks4.pop()
22 maksymalna = 0
23 for i in range(3):
24     for j in range(i + 1, 4):
25         if (not (maks4[i][1]==0 and maks4[j][1]==n-1)) and
            (not (maks4[i][1]==n-1 and maks4[j][1]==0)) and
            abs(maks4[i][1] - maks4[j][1]) > 1 and
            maks4[i][0] + maks4[j][0] > maksymalna:
26             maksymalna = maks4[i][0] + maks4[j][0]
27 print(suma-maksymalna)
```



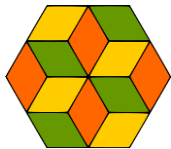
Rozwiązanie należy przetestować najpierw na przykładach z treści zadania, następnie na większych danych. Podczas konkursu zdanie było testowane na następujących grupach testów.

- 4 -



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

	898765432345623456789876876 543234567898765432345678987 654323456234567898761187654 323456789876543234567898765 432345623456789876876543232 345678987654323456234567898 768765432345678987654323456 789876543234562345678987687 654323456789876543234567898 765432345623456711898768765 432345678987654323456789876 543234562345678987687654323 456789876543234567898765432 345623456789876876543234567 898765432345678987654323456 234567898768765432323456789 876543234562345678987687654 323456789876543234567898765 432345623456789876876543234 567891187654323456789876543 234562345678987687654323456 789876543234567898765432345 623456789876876543234567898 765432345678987654323456234 567118987687654323456789876 543234567898765432345623456 789876876543234567891187654 323456234567898768765		
V	6 10000 98 2 1 1 99	199	trzy największe obok siebie, dwie na brzegu
	100 993456787987654323456789876 543234567898765432345623456 789876823456789876543234567 8987654323456789878100110	723	
	1000 999345678345678345678345678 345678118876543234567898765 432345678987654323456234567 898768765432345678987654323 456789876543234562345678987 687654323456789876543234567 898765432345623456789876876 543234567898765432345678987 654323456234567898768765432 345678987654323456789876543 234562345678987611876543234 567898765432345678987654323	8051	



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

4 5 6 2 3 4 5 6 7 8 9 8 7 6 8 7 6 5 4 3 2 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 2 3 4 5 6 7 8 9 8 7 6 8 7 6 5 4 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 2 3 4 5 6 7 8 9 8 7 6 8 7 6 5 4 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 2 3 4 5 6 7 1 1 8 9 8 7 6 8 7 6 5 4 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 2 3 4 5 6 7 8 9 8 7 6 8 7 6 5 4 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 2 3 4 5 6 7 8 9 8 7 6 8 7 6 5 4 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 2 3 4 5 6 7 8 9 8 7 6 8 7 6 5 4 3 2 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 2 3 4 5 6 7 8 9 8 7 6 8 7 6 5 4 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 2 3 4 5 6 7 8 9 8 7 6 8 7 6 5 4 3 2 3 4 5 6 7 8 9 1 1 8 7 6 5 4 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 2 3 4 5 6 7 8 9 8 7 6 8 7 6 5 4 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 2 3 4 5 6 7 8 9 8 7 6 8 7 6 5 4 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 2 3 4 5 6 7 1 1 8 9 8 7 6 8 7 6 5 4 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 2 3 4 5 6 7 8 9 8 7 6 8 7 6 5 4 3 2 3 4 5 6 7 8 9 1 1 8 7 6 5 4 3 2 3 4 5 6 2 3 4 5 6 7 8 9 8 7 6 8 7 6 5 8 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 3 4 5 6 7 8 9 1 1 8 7 6 5 4 3 2 3 4 5 6 7 8 9 8 0 1 0 0 0 0		
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--