

Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Zadanie Palindromy – LOGIA (2019/2020), etap 2

Treść zadania

Palindrom to napis, który czytany od lewej do prawej i od prawej do lewej jest identyczny, na przykład: *oko*, *kajak*, *abba*. Napis możemy rozszerzyć poprzez dodawanie liter na początku albo na końcu. Nie można dodawać liter w środku ani jednocześnie z obu stron.

Napisz funkcję **roz(napis)**, której parametrem jest niepusty napis złożony z małych liter alfabetu łacińskiego o długości nie większej niż 1 000. Wynikiem jest napis rozszerzony o minimalną liczbę znaków tak, aby powstał palindrom. Jeśli takich napisów jest więcej niż 1, wynikiem funkcji powinien być napis pierwszy w porządku alfabetycznym. Przy czym rozszerzenie może być o 0 znaków, gdy parametr jest już palindromem.

Przykłady:

Wynikiem **roz("ko")** jest **"kok"** (możliwe palindromy to "oko" i "kok").

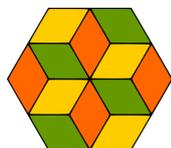
Wynikiem **roz("abba")** jest **"abba"**.

Wynikiem **roz("abbaca")** jest **"acabbaca"**.

Omówienie rozwiązania

Rozwiązanie polega na sprawdzaniu, czy dane słowo jest palindromem i w przypadku negatywnej odpowiedzi rozszerzanie napisu o kolejną literę. Sprawdzanie czy napis jest palindromem, realizujemy poprzez porównanie napisu oraz napisu odwróconego. Jeśli są identyczne, napis jest palindromem, w przeciwnym przypadku nie jest.

W pętli zaczynamy od kroku 0, czyli od sprawdzania, czy dane słowo jest palindromem. W i-tym kroku algorytmu rozszerzamy napis o i-tą literę, a dokładniej rozszerzając z tyłu bierzemy i-liter początkowych i dopisujemy je w odwrotnej kolejności na koniec. Następnie sprawdzamy, czy tak powstały napis jest palindromem. Podobnie postępujemy, rozpatrując dopisywanie od przodu, z tym że rozpatrujemy odwrócony napis. Na koniec i-tego kroku pętli sprawdzamy, czy otrzymaliśmy palindrom: tylko w pierwszym przypadku – dopisanie liter na koniec, tylko w drugim przypadku – dopisanie liter na początek, czy w obu. Jeśli odpowiedź jest twierdząca, to otrzymany palindrom jest on wynikiem funkcji. Przy czym, gdy w danym obrocie pętli wygenerowane zostały dwa palindromy, to przekazujemy jako wynik wcześniejszy leksykograficznie. Jeśli nie znaleźliśmy palindromu, postępowanie kontynuujemy tak długo, aż nie otrzymamy palindromu. Górnym ograniczeniem na długość słowa jest dwukrotna długość wyjściowego słowa pomniejszona o 1.



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Rozwiązanie w języku Python

W języku Python istnieje możliwość wyodrębnienia fragmentu napisu poprzez notacją nawiasową. Zapis `napis[a:b:c]` oznacza, że ze zmiennej typu napis wybieramy fragment od elementu `a` do `b` co `c` elementów. Jeśli wybieramy od początku napisu, to pomijamy `a`, jeśli do końca to pomijamy `b`. Wobec tego zapis `napis[::-1]` w istocie powoduje odwrócenie słowa.

```
1. def czy_pali(napis):
2.     return napis==napis[::-1]
3.
4. def rozszerz(napis,ile):
5.     return napis + (napis[:ile])[::-1]
6.
7.
8. def roz(napis):
9.     for i in range(len(napis)):
10.        ile = 0
11.        x = rozszerz(napis,i)
12.        y = rozszerz(napis[::-1],i)
13.        if czy_pali(x):
14.            ile += 1
15.        if czy_pali(y):
16.            ile += 2
17.        if ile == 1:
18.            return x
19.        if ile == 2:
20.            return y
21.        if ile == 3:
22.            if x < y:
23.                return x
24.            else:
25.                return y
26.    return -1
```

W powyższym zapisie użyto zmiennej pomocniczej `ile`, która wskazuje jakie palindromy otrzymamy w `i`-tym obrocie pętli. Jeśli zmienna `ile` ma wartość 0 – to nie otrzymaliśmy palindromu, jeśli 1 – otrzymano palindrom przez dopisanie liter na końcu, jeśli 2 – na początku. Gdy uzyskano oba palindromy, to 3.

Testy

Testy obejmują różne przypadki – gdy dopisujemy jedną lub więcej liter na końcu, na początku lub w ogóle nie dopisujemy – dana jest palindromem. W Pythonie jest zdefiniowana operacja mnożenia liczby przez napis. Zapis `7*"kajak"` oznacza powtórzenie 7 razy napisu "kajak", czyli

'kajakkajakkajakkajakkajakkajakkajak'.

Przykładowe testy:

`roz("defdefdeffedfed")` → `defdefdeffedfedfed`

`roz("zzzxyxyxy")` → `zzzxyxyxyzzz`

`roz("uuuughturguuuu")` → `uuuughturguuuuugruthguuuu`

