

Kurs Pythona v0.2

Lista zadaniowa V

Wojciech Adamiec

14 grudnia 2022

Spis treści

1	Najdłuższe słowo	2
2	Unikaty	3
3	Kropkowy obrazek	4
4	Wyrazy odwrotne	5
5	Wieża Hanoi	6

1 Najdłuższe słowo



W repozytorium do listy znajduje się korpus słów języka polskiego `words.txt`. Napisz procedurę `longest_word()`, która dla wspomnianego korpusu słów zwróci (dowolne) najdłuższe słowo.

2 Unikaty



Napisz procedurę `unique(data)`, która dla listy wejściowej `data` zawierającej potencjalnie duplikaty elementów zwróci listę wynikową bez żadnych duplikatów.

Zadbaj o to, aby nie zmienić kolejności elementów na liście. Jeśli jakiś element pojawia się wielokrotnie to w liście wynikowej powinno pojawić się tylko pierwsze jego wystąpienie.

Przykładowe wywołania funkcji:

```
unique([1, 1, 1]) -> [1]
unique([1, 2, 3]) -> [1, 2, 3]
unique([1, 2, 3, 1]) -> [1, 2, 3]
unique([1, 2, 3, 1, 2, 3, 4]) -> [1, 2, 3, 4]
```

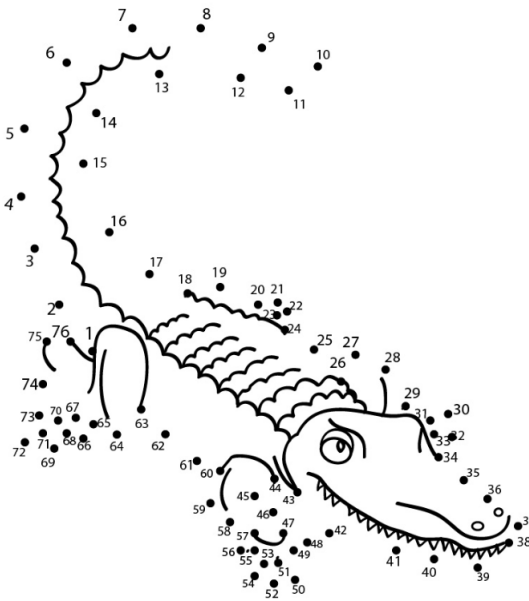
Wskazówka: Wykorzystaj w rozwiązaniu zbiory.

3 Kropkowy obrazek



W pliku `dots.txt` znajdują się dane ze współrzędnymi punktów. Napisz program, który narysuje obrazek na podstawie danych z pliku poprzez połączenie owych punktów.

Zadanie jest programistycznym odpowiednikiem powszechnie znanych obrazków z serii *connect the dots*. Przykładowy obrazek poniżej:



4 Wyrazy odwrotne



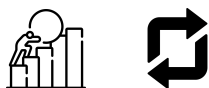
Parę wyrazów nazwiemy wzajemnie odwrotnymi, jeżeli pierwszy z nich jest równy drugiemu przeczytanemu wspak. Przykładową parą słów odwrotnych jest para: **zakop** oraz **pokaz**.

Napisz procedurę `reversed_words()`, która dla korpusu słów języka polskiego `words.txt` zwróci w postaci listy par (krotek dwuelementowych) wszystkie wyrazy odwrotne.

Postaraj się zoptymalizować swoje rozwiązanie tak, aby uniknąć pętli w pętli (do generowania wszystkich par słów).

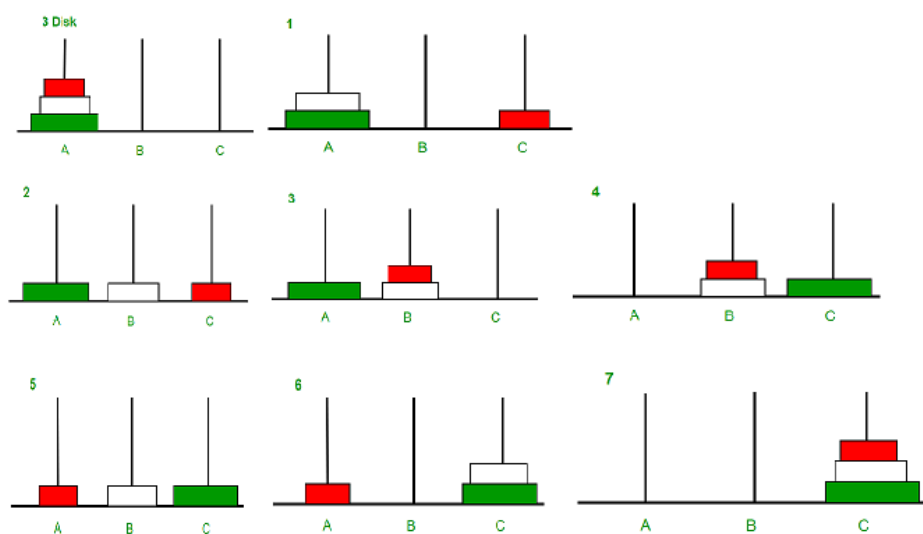
Uwaga! W liście wynikowej każda para słów odwrotnych powinna się pojawić dokładnie jeden raz. W ramach każdej pary obowiązuje kolejność alfabetyczna (poprawna jest para ("`fgh`", "`hgf`"), ale nie ("`hgf`", "`fgh`"). Na samej liście wynikowej również obowiązuje kolejność alfabetyczna.

5 Wieże Hanoi



Napisz rekurencyjną procedurę `tower_of_hanoi(n, source, dest, aux)`, która dla `n` dyskowej wieży `source` (gdzie `source` jest nazwą wieży) wypisze ciąg kolejnych przesunięć, które należy wykonać, aby za pomocą pomocniczej wieży `aux` przenieść wszystkie dyski do wieży `dest`.

Szczegóły dotyczące reguł wież Hanoi można znaleźć na przykład na stronie [Wikipedii](#). Poniżej znajduje się przykładowy schemat rozwiązania wieży Hanoi dla danych: (`n: 3`, `source: A`, `dest: C`, `aux: B`).



Przykładowe wywołanie programu `tower_of_hanoi(4, "A", "B", "C")` powinno na terminal wypisać:

```
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 4 from A to B
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 3 from C to B
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
```