

Algorytm i Struktury Danych

Wykład 4

Sortowanie u cznie liniowym - sortowanie
przez zliczanie

tabela zawiera liczby naturalne
ze zbioru $\{0, \dots, k-1\}$

def counting_sort(A, k):

n = len(A)

B = [None] * n

C = [0] * k

for x in A: C[x] += 1

for i in range(1, k): C[i] += C[i-1]

for i in range(n-1, -1, -1):

B[C[A[i]]-1] = A[i]

C[A[i]] -= 1

for i in range(n):

A[i] = B[i]

każde C[i] zawiera liczbę
wartości z A, które są $\leq i$

k = 5

	0	1	2	3	4	5	6	7
A:	1	2'	0'	2''	3	0''	4	0'''

←

	0	1	2	3	4
C:	3	1	2	1	1

↓

	0	1	2	3	4
C:	3	4	6	7	8

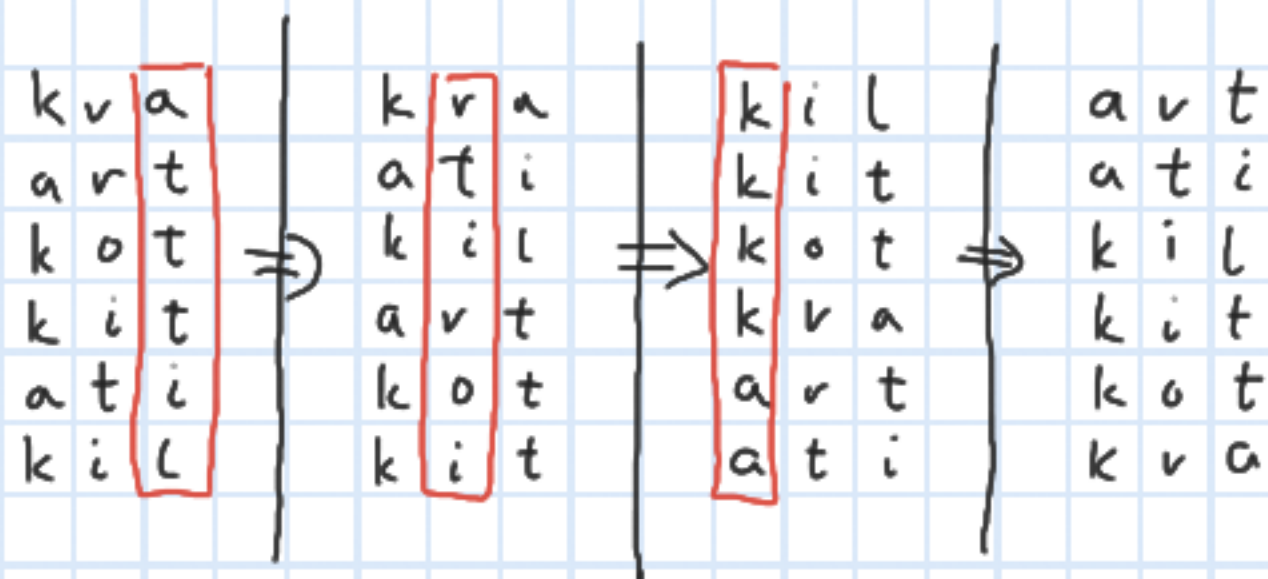
	0	1	2	3	4	5	6	7
B	0'	0''	0'''	1	2'	2''	3	4

	0	1	2	3	4
C	0	3	4	6	7

złożoność
counting sort

$O(n+k)$

Sortowanie pozycyjne - Radix sort



Sortowanie kuletkowe - Bucket Sort

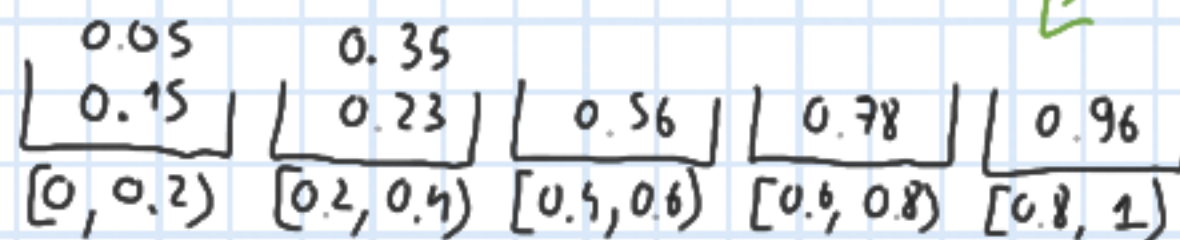
- sortowanie liczb wymiarowych, ze zbiorem $[0, 1)$
 wygenerowane z rozkładu jednostajnego

0.15, 0.23, 0.78, 0.56, 0.35, 0.96, 0.05

kuletków powinno być

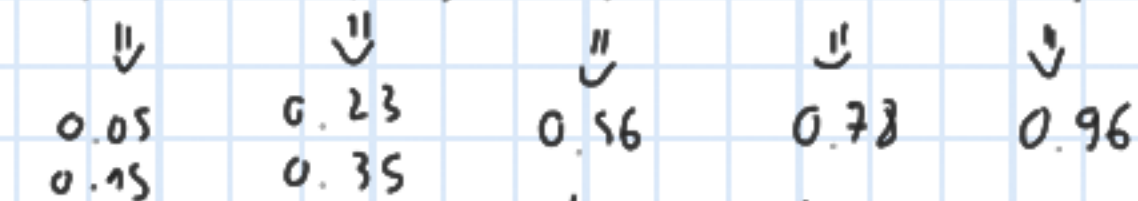
(n)

rozmian danych
do sortowania



sortowanie kuletek
np. przez ustawianie

jeśli kuletek drugi,
to QuickSort



0.05, 0.15, 0.23, 0.35, 0.56, 0.78, 0.96

jeśli kuletek jest n , to $A[i]$
 umieszczamy w kuletkach $[n \cdot A[i]]$

Abstrakcyjne struktury danych

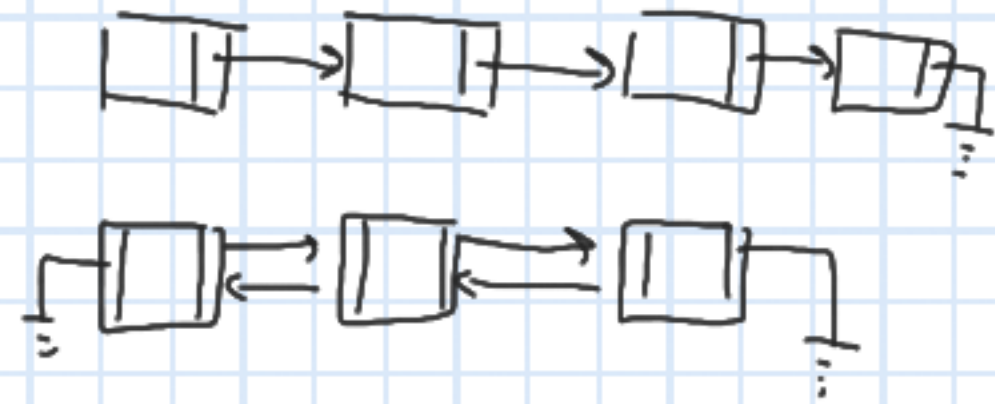
↳ "kontrakt" między strukturą a jej
 użytkownikiem — zbiór operacji
 ↳ realizacja "fizyczna"

Tablica



"Coś, co pozwala odwoływać się do
 komórek po ich numerach"

Lista jedno/dwukierunkowa

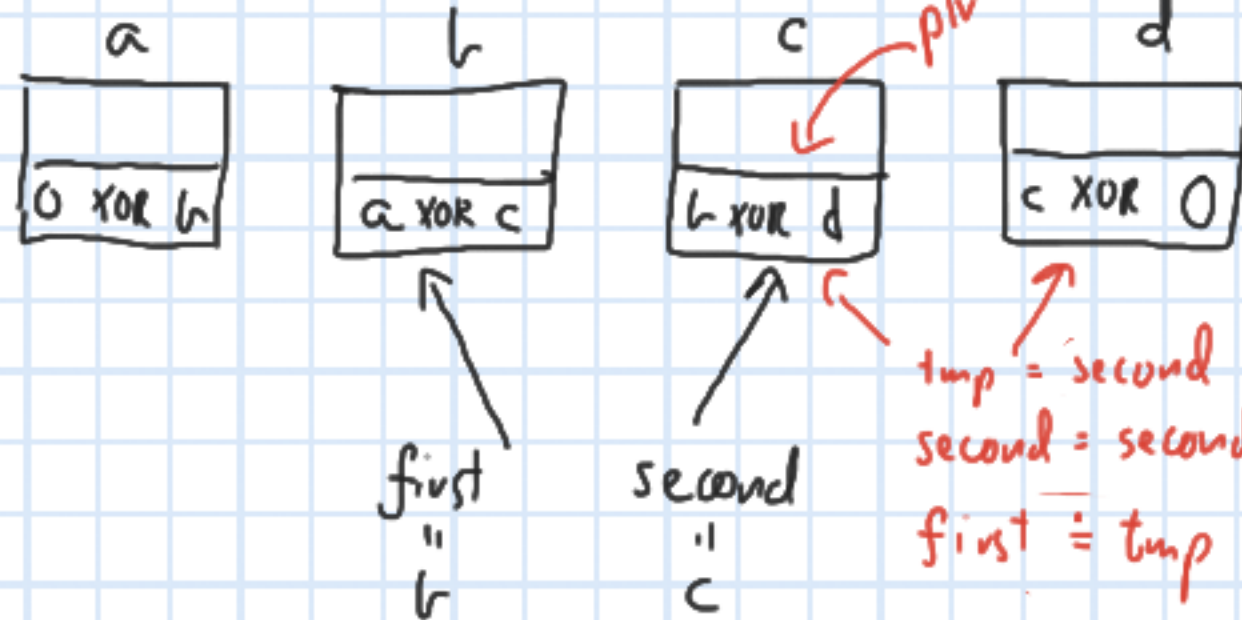


"Coś, co pozwala poruszać się po kolei po elementach w jednym lub dwie strony i upinać lub usuwać elementy"

```
class DNode:
    def __init__(self, val):
        self.prev = None
        self.next = None
        self.val = val
```

↑
wzrost listy dwukierunkowej

Lista dwukierunkowa "na jednym wskazniku"



tmp = second
second = second.ptv XOR first
first = tmp

$$\begin{array}{r} 1100110 \\ 0101100 \\ \hline \text{XOR} \\ 1001010 \\ \\ 1001010 \\ 0101100 \\ \hline \text{XOR} \\ 1100110 \end{array}$$

Stos

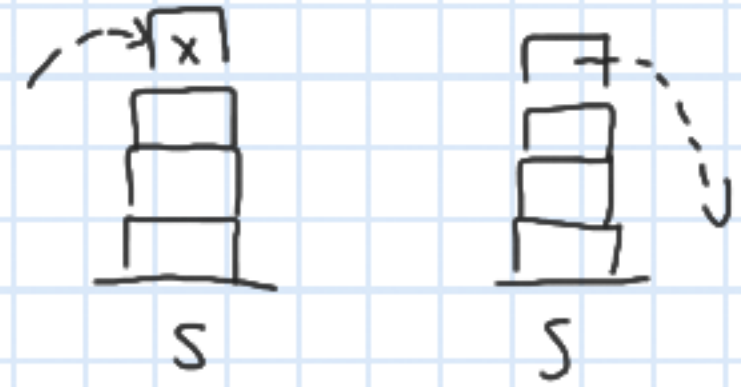
"Coś, co pozwala na odłożenie na wienholde i zdejmowanie z niego"

S - stos

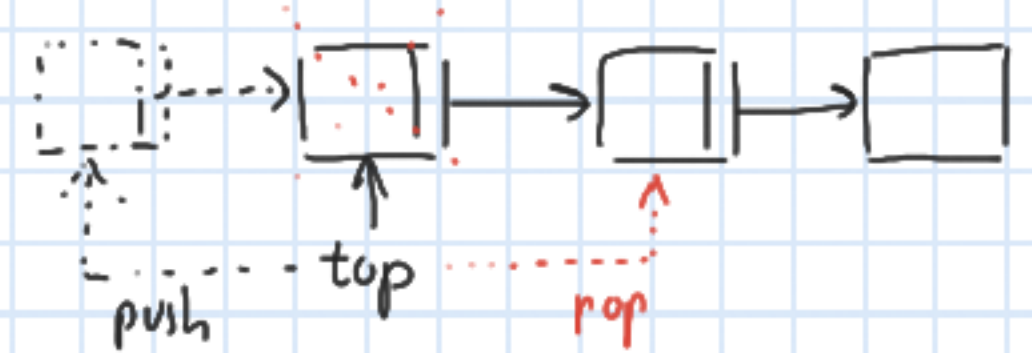
S.push(x)

S.pop()

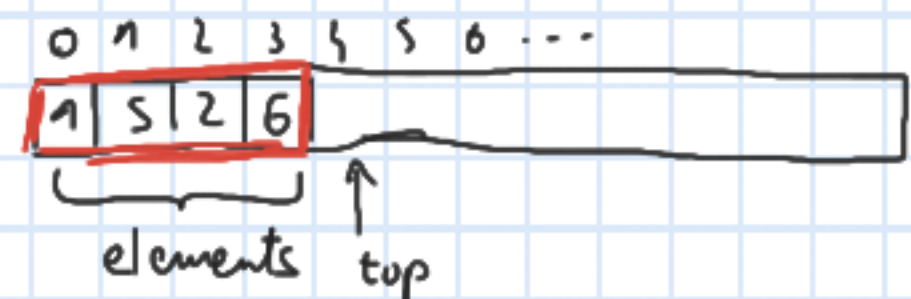
S.isempty()



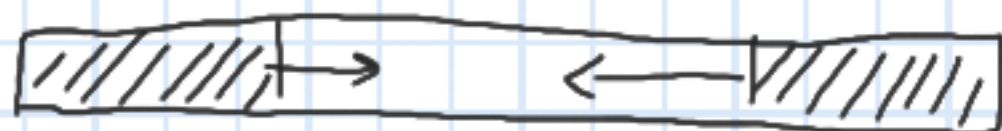
Implementacja listowa



Implementacja tablicowa



Dva stosy?

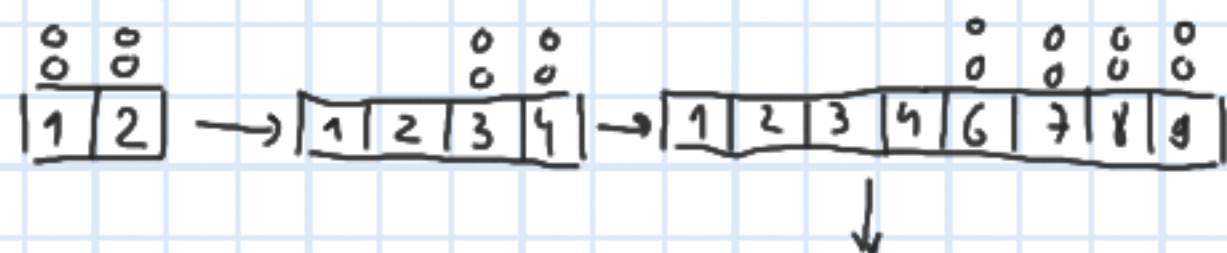


Co zrobić, gdy stos "tablicowy" się przepelni?

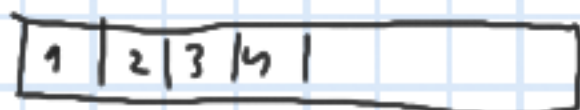
↳ zaalokować nową, 2x większą tablicę i tam skopiować stos

- push - 3 zł

- pop - 1 zł



Mogę też zmniejszyć tablicę, ale np. dopiero jak wypełniona powyżej 1/4 pojemności



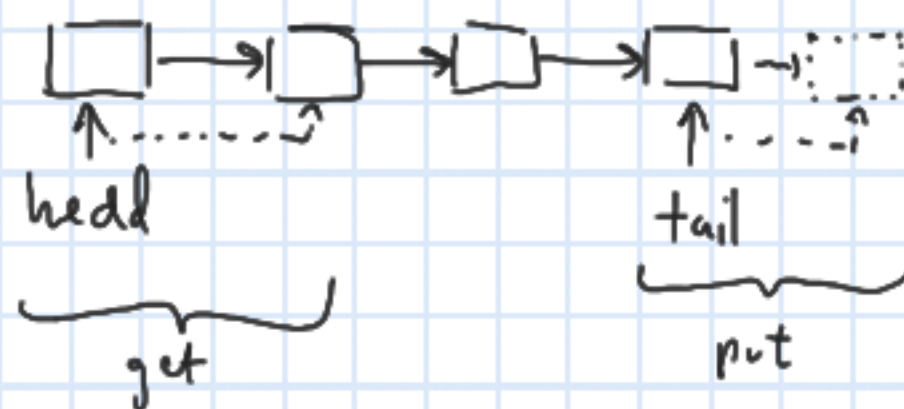
Kolejka

"Coś, co pozwala dodawać elementy na koniec i zaliczać je z początku"

- enqueue (put)

- deque (get)

Implementacja listowa



Implementacja tablicowa (przebiega kolejka)

