

Algorytmy i struktury danych

Laboratorium 2

mgr inż. Bartłomiej Kizielewicz, mgr inż. Andrii Shekhovtsov

13 listopada 2023

1 Zasady oceniania

Ocena za laboratorium zależy od liczby dobrze zrobionych zadań domowych, skala ocen jest pokazana w Tabeli 1. W niektórych przypadkach mogą być uwzględnione zadania rozwiązane częściowo.

Tabela 1: Skala ocen.	
Liczba zrobionych zadań domowych	Ocena
1	3.0
2	4.0
3	4.5
4	5.0

Wykonane zadania proszę przysłać za pośrednictwem platformy moodle w postaci pliku tekstowego z kodem w Python (plik z rozszerzeniem **.py**). Proszę wrzucić kod ze wszystkich zadań do jednego pliku, rozdzielając poszczególne zadania za pomocą komentarzy.

UWAGA: Termin oddania zadania jest ustawiony w systemie moodle. W przypadku nie oddania zadania w terminie, uzyskana ocena będzie zmniejszana o 0,5 za każdy zaczęty tydzień opóźnienia. Zadania oddawane później niż miesiąc po terminie ustawionym na moodle są oddawane i rozliczane w trybie indywidualnym na zajęciach lub po umówieniu się z prowadzącym.

UWAGA: W przypadku wysłania zadania w formie niezgodnej z opisem w instrukcji prowadzący zastrzega prawo do wystawienia oceny negatywnej za taką pracę. Przykład: wysłanie **.zip** lub **.pdf** tam, gdzie był wymagany plik tekstowy z rozszerzeniem **.py**.

2 Zadania do wykonania na zajęciach

1. Zaimplementować algorytm wyszukiwania zwykłego w liście (nie ma założenia że lista zawiera wartości posortowane). Przetestować działanie tego algorytmu i oszacować jego złożoność obliczeniową. Krótko (1-2 zdania) opisać dlaczego ten algorytm ma taką złożoność obliczeniową.
2. Zaimplementować **iteracyjny** algorytm wyszukiwania binarnego (binary search). Przetestować jego działanie i oszacować złożoność obliczeniową tego algorytmu. Krótko (1-2 zdania) opisać dlaczego ten algorytm ma taką złożoność obliczeniową.
3. Zaimplementować algorytm który sprawdzi czy w dwóch zadanych listach o różnych długościach występuje taka sama liczba. Poniżej jest podany przykład działania takiego algorytmu. Następnie należy oszacować złożoność obliczeniową tego algorytmu i krótko (1-2 zdania) opisać dlaczego ten algorytm ma taką złożoność obliczeniową.

Przykład:

```
1 >>> a = [1, 2, 3]
2 >>> b = [4, 5]
3 >>> magic(a, b)
4 False
```

```

5 |
6 | >>> a = [1, 2, 3]
7 | >>> b = [3, 4]
8 | >>> magic(a, b)
9 | True

```

4. Zaimplementować algorytm sortowania bąbelkowego (bubble sort). Przetestować jego działanie i oszacować złożoność obliczeniową tego algorytmu. Krótko (1-2 zdania) opisać dlaczego ten algorytm ma taką złożoność obliczeniową.

3 Zadania do wykonania w domu

1. Zaimplementować wyszukiwanie największej liczby w nieposortowanej liście liczb. Przetestować jego działanie i oszacować złożoność obliczeniową tego algorytmu. Krótko (1-2 zdania) opisać dlaczego ten algorytm ma taką złożoność obliczeniową.
2. Zaimplementować algorytm sortowania przez wstawianie (insertion sort). Przetestować jego działanie i oszacować złożoność obliczeniową tego algorytmu. Krótko (1-2 zdania) opisać dlaczego ten algorytm ma taką złożoność obliczeniową.
3. Zaimplementować opisany poniżej algorytm i oszacować jego złożoność obliczeniową. Krótko (1-2 zdania) opisać dlaczego ten algorytm ma taką złożoność obliczeniową.

Algorytm: na wejściu funkcja dostaje dwa ciągi znaków o różnych długościach $s1$ i $s2$. Algorytm ma zliczyć ile razy w ciągu $s1$ występują znaki zawierające się w ciągu $s2$. W poniższym przykładzie zliczamy ile samogłosek występuje w podanym ciągu.

Przykład:

```

1 | >>> s1 = 'ała ma kota'
2 | >>> s2 = 'aoiuye'
3 | >>> wynik = magic(s1, s2)
4 | >>> print(wynik)
5 | 5

```

Podpowiedź: Jako dane wejściowe mamy dwa ciągi znaków $s1$ o długości N oraz $s2$ o długości M .

4. Zaimplementować algorytm wyznaczający medianę w posortowanej liście liczb. Oszacować złożoność obliczeniową tego algorytmu oraz krótko (1-2 zdania) opisać dlaczego ten algorytm ma taką złożoność.

Podpowiedź: Medianę posortowanej listy o długości n obliczamy jako wartość w środku listy (jeżeli n jest nieparzyste). Jeśli natomiast n jest parzyste, wynikiem jest średnia arytmetyczna między dwiema środkowymi elementami listy.