

# Sprawozdanie z projektu pierwszego - Struktury Baz Danych

Piotr Sieński

4 grudnia 2022

## 1 Opis projektu

### 1.1 Typ rekordów

Wylosowanym typem rekordów są walce o danym promieniu podstawy i wysokości sortowane według objętości.

### 1.2 Zastosowana metoda

Do realizacji zadania wykorzystana została metoda sortowania polifazowego (fibonacciego) na 3 taśmach.

### 1.3 Specyfikacja formatu pliku testowego

Serializacja rekordów przeprowadzana jest za pomocą modułu json języka Python. Zakłada się że w każdej linii pliku testowego znajduje się opis jednego rekordu w formacie json, tzn.

```
{"radius": promień podstawy, "height": wysokość}
```

gdzie promień podstawy i wysokość są liczbami zmiennoprzecinkowymi. Przykładowa zawartość pliku zawierającego dwa rekordy o promieniach podstawy 1 i wysokościach 1 i 3 :

```
{"radius": 1, "height": 1}  
{"radius": 1, "height": 3}
```

## 2 Sposób prezentacji wyników działania programu

### 2.1 Uruchomienie programu

W celu uruchomienia programu należy wykonać poniższą komendę w folderze, gdzie znajduje się plik main.py

```
python main.py
```

### 2.2 Opcje generowania rekordów

Istnieją 3 możliwości na podanie danych wejściowych do programu, po starcie należy wybrać jedną z nich po pojawieniu się następującego komunikatu:

```
Type "lf" to load whole file , "gr" to generate file with random records ,  
"lr" to read records from STDIN "q" to quit
```

#### 2.2.1 Ładowanie wcześniej przygotowanego pliku

Po wybraniu opcji "lf" czyli ładowania przygotowanego wcześniej pliku należy podać ścieżkę do pliku zawierającego rekordy podane w odpowiednim formacie (opisanym w punkcie 1.3)

## 2.2.2 Generowanie losowego pliku

Po wybraniu opcji "gr" czyli generowania pliku z losowymi rekordami należy podać liczbę rekordów do wylosowania. Wylosowane wartości pól rekordów należą do przedziału (0, 1)

## 2.2.3 Generowanie nowego pliku rekord po rekordzie

Po wybraniu opcji "lr" wyświetli się następujący komunikat :

Type "l" to load single record from STDIN, "r" to add random\_record,  
"s" to save, "q" to quit

Dopóki użytkownik nie wybierze opcji "s" możliwe będzie wybranie opcji "r", czyli dodania pojedynczego rekordu o losowej wartości oraz opcji "l", czyli dodanie rekordu o podanej wartości.

## 2.3 Etap sortowania

Po wygenerowaniu lub załadowaniu pliku z rekordami zostaną one wyświetlone oraz możliwe będzie wybranie czy między fazami sortowania wyświetlane będą informacje na temat liczby odczytów / zapisów i zawartości taśm. Przykładowe wyjście przed rozpoczęciem sortowania:

LOADED FILE:

Right circular cylinder — base radius: 0.66, height: 0.33, volume: 0.46  
Right circular cylinder — base radius: 0.65, height: 0.95, volume: 1.25  
Right circular cylinder — base radius: 0.70, height: 0.76, volume: 1.16  
Run in verbose mode ? (y/n)

Jeśli wybrana zostanie opcja verbose = n, od razu zostanie wyświetlona liczba serii na pliku wejściowym, liczba faz sortowania, liczba odczytów (łącznie z odczytami z pliku wejściowego) i zapisów oraz posortowany plik:

Initial runs: 2  
File sorted in 1 phases  
WRITES: 3 + READS: 3 = 6  
Right circular cylinder — base radius: 0.66, height: 0.33, volume: 0.46  
Right circular cylinder — base radius: 0.70, height: 0.76, volume: 1.16  
Right circular cylinder — base radius: 0.65, height: 0.95, volume: 1.25

Jeśli wybrana zostanie opcja verbose = t, dodatkowo po każdej z faz sortowania (włącznie z etapem dystrybucji) wyświetlona zostanie zawartość każdej z taśm wraz z liczbą serii na taśmie i liczbą odczytów / zapisów z / do pliku przypisanego do taśmy

AFTER PHASE: 0

Tape0:

RUNS: 1, DUMMY RUNS: 0

WRITES: 1, READS: 0

Right circular cylinder — base radius: 0.66, height: 0.33, volume: 0.46  
Right circular cylinder — base radius: 0.70, height: 0.76, volume: 1.16  
Right circular cylinder — base radius: 0.65, height: 0.95, volume: 1.25

Tape1:

RUNS: 0, DUMMY RUNS: 0

WRITES: 1, READS: 1

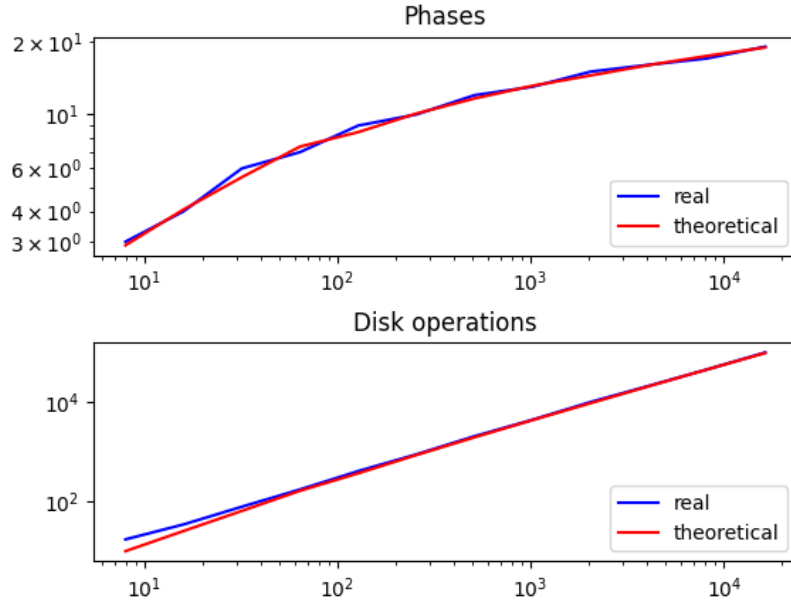
Tape2:

RUNS: 0, DUMMY RUNS: 0

WRITES: 1, READS: 1

### 3 Eksperyment

Eksperyment polega na porównaniu wartości teoretycznych liczby faz sortowania i ilości operacji dyskowych z wartościami uzyskanymi podczas sortowania. Wartość teoretyczna liczby faz  $p$  w zależności od liczby serii w pliku wejściowym  $r$  można wyznaczyć jako  $p = 1.45 \cdot \log_2 r$ , a ilość operacji dyskowych  $o$  w zależności od liczby rekordów  $N$ , liczby serii  $r$  i ilości rekordów na blok pamięci  $b$  jako  $o = \frac{2N}{b}(1.04 \cdot \log_2 r + 1)$ . Obliczone według powyższych wzorów wartości zestawione zostały z realnymi wartościami dla losowo wygenerowanych plików zawierających  $n$  rekordów dla  $n \in \{8, 16, 32, \dots, 8192\}$  i  $b = 5$ . Poniższy wykres i tabela przedstawiają wyniki eksperymentu.



$N$	8	16	32	64	128	256	512	1024	2048	4096	8192
$O_R$	17	34	77	167	406	874	2002	4319	9799	20761	44281
$O_T$	9.85	25.08	63.48	154.93	364.43	853.75	1895.24	4237.42	9337.67	20385.29	44216.66
$P_R$	3	4	6	7	9	10	12	13	15	16	17
$P_T$	2.9	4.07	5.52	7.04	8.52	10.23	11.50	13.02	14.49	15.95	17.41

$O_R, O_T$  - liczba operacji rzeczywista i teoretyczna,  $P_R, P_T$  - ilość faz,  $N$  - ilość rekordów

Jak widać wartości są bardzo podobne i dla większych  $N$  różnice są coraz mniejsze. Lekkie rozbieżności mogą wynikać z faktu, że przybliżenie bierze pod uwagę sklejające się serie, które mogą występować z różną częstotliwością.