

Politechnika Wrocławska

Wydział Informatyki i Telekomunikacji

Kierunek: **Informatyka Algorytmiczna (INA)**

Specjalność: **Informatyka po angielsku (IAG)**

PRACA DYPLOMOWA MAGISTERSKA

Szkice danych dla algorytmów strumieniowych

Anna Adamowicz

Opiekun pracy:
prof. dr hab. Krzysztof Kowalski

Słowa kluczowe: *strumień danych, Big Data, zliczanie elementów, rozkład jednostajny*

Wrocław 2026

Streszczenie

Tutaj umieść zwięzłe streszczenie swojej pracy dyplomowej. Powinno ono zawierać opis celu pracy, zastosowane metody oraz uzyskane wyniki.

Oba streszczenia, po polsku i po angielsku, powinny zmieścić się na jednej stronie.

Abstract

Place a concise summary of your thesis here. It should include a description of the purpose of the work, the methods used and the results obtained.

Spis treści

1	Wstęp	1
1.1	Cel i struktura pracy	1
1.2	Wykorzystanie AI	2
1.3	Wprowadzenie	2
1.4	Podstawy teoretyczne	2
2	O szablonie	3
2.1	Definicje użytkownika	4
2.2	Ilustracje	4
2.3	Wzory matematyczne	4
2.4	Cytowania	5
2.5	Tabele	6
2.6	Kody procedur	6
2.7	Akronimy	7

Lista skrótów

DRY Don't repeat yourself	7
KISS Keep It Simple, Stupid	7

Rozdział 1

Wstęp

Tutaj napisz krótkie wprowadzenie do swojej pracy. Możesz omówić podstawowe algorytmy związane z celem swojej pracy. Możesz podać krótki rys historyczny. Pamiętaj jednak o zwięźłości tekstu i o unikaniu nic nie wnoszących ogólników. Oto przykładowy początek takiego wprowadzenia:

Problem algorytmów wyspecjalizowanych do analizy danych strumieniowych wzbudził duże zainteresowanie badaczy na początku XX wieku. Jednym z pierwszych takich algorytmów służył do wyboru losowego elementu zgodnie z rozkładem jednostajnym z okna złożonego z ostatnich N elementów strumienia, gdzie N jest ustaloną liczbą. ...

1.1. Cel i struktura pracy

Postaraj się tutaj opisać możliwie zwięźle i precyzyjnie cel pracy. Oto przykład początku tej sekcji pracy:

W pracy tej będziemy zajmować się problemem wyznaczania k - najczęściej występujących elementów w oknie złożonym z N ostatnich elementów strumienia danych. ...

Struktura pracy

Tutaj opisz krótko, po kolei, wszystkie rozdziały swojej pracy. Oto przykład:

Rozdział pierwszy zaczynamy (patrz 1.3) od opisu wymagań stawianych algorytmom strumieniowym (główne wymaganie to wymaganie użycia logarytmicznej pamięci względem rozmiaru strumienia), opisujemy pojęcie przesuwne okna strumienia danych, omawiamy algorytm Bravermana,

W rozdziale drugim omawiamy ...

...

Zastosowane technologie

Tutaj opisz, z jakich języków programowania oraz z jakich kompilatorów korzystałeś, w jakim środowisku uruchamiałeś oraz testowałeś budowane oprogramowanie. Wskaż również najważniejsze biblioteki wykorzystane w zbudowanych programach.

1.2. Wykorzystanie AI

Jeśli podczas realizacji pracy wykorzystywano współczesne narzędzia AI, to w tej części należy je opisać, czyli napisać, jakie narzędzia AI były wykorzystywane, do czego były używane oraz których miejsc pracy to dotyczy. **Tej części pracy nie możesz pominąć**, aczkolwiek możesz ją umieścić w innym miejscu pracy. Koniecznie zapoznaj się z “Wytycznymi dotyczącymi zasad wykorzystywania AI”, które znajdują się na stronach naszego Wydziału (patrz [1]).

Jeśli AI było wykorzystywane do generowania kodów, to również o tym tutaj napisz, a w kodach programu (w komentarzach) zaznacz te fragmenty, w które ingerowała AI. Oto przykład:

W pracy korzystano z asystenta Gemini 3.0 firmy Google w wersji Pro do pomocy w przekształceniu kodów z języka programowania C na język programowania Haskell (patrz, między innymi, listing 2.2) oraz do pomocy w dokumentowaniu kodów wszystkich zbudowanych w trakcie realizacji pracy modułów.

Jeśli podczas realizacji pracy nie była wykorzystywana AI, to w sekcji tej umieść mniej więcej takie zdanie: *"W pracy nie korzystano ze Sztucznej Inteligencji"*.

1.3. Wprowadzenie

Tutaj możesz opisać podstawowe pojęcia, terminy i oznaczenia, którymi posługujesz się w pracy.

1.4. Podstawy teoretyczne

Możesz podać podstawowe wyniki, z których korzystasz w pracy. Mogą to być odwołania do literatury - w rozdziale 2.4 omówiono sposób formatowania bibliografii oraz metody odwoływania się w pracy do konkretnych pozycji (`\cite{...}`, `\citeauthor{...}`, ...)

Rozdział 2

O szablonie

W rozdziale tym znajduje się kilka wskazówek, które mogą Ci się przydać podczas redagowania pracy oraz przykłady użycia pakietów systemu Latex, które są dołączone do tego szablonu. Szablon ten bazuje na klasycznej klasie `report`. Najważniejszym składnikiem tego szablonu jest plik `settings.tex`. Oto kilka szczegółów:

1. Wszystkie pliki pracy powinny być zapisane w formacie utf-8.
2. Posługujemy się czcionkami z rodziny `newtxtext` oraz `newtxmath`. Są to nowoczesne wersje fontów w stylu Times. Głównym fontem dokumentu jest czcionka szeryfowa. Strona tytułowa jest wyświetlana wariantem bezszeryfowym.
3. Do wyświetlania bibliografii posługujemy się pakietem `biblatex`, a do przetwarzania bibliografii programem `biber`. Jest to nowoczesna wersją starego programu `bibtex`. Przed użyciem tego szablonu dowiedz się, jak skonfigurować środowisko, które używasz do składania plików Latex’a, tak aby używało ono `biber`’a. Jeśli masz z tym problem, spróbuj z linii poleceń uruchomić polecenie `biber plik`, gdzie `plik` jest nazwą głównego pliku projektu bez rozszerzenia. Jeśli korzystasz ze środowiska [Overleaf](#), to ono samo powinno wykryć stosowanie bibera.
4. Jeśli korzystasz z lokalnych środowisk do składania plików Latex’a (np. TeXnicCenter), to pierwsza kompilacja z użyciem tego szablonu może zająć trochę czasu, gdyż środowisko może doinstalowywać potrzebne pakiety.
5. Szablon importuje pakiet `listings` do wyświetlania “listingów” programów oraz pakiety `algorithm` i `algpseudocode` do wyświetlania pseudokodów.
6. Szablon importuje pakiet `acro` do operowania synonimami. Jeśli nie stosujesz synonimów, to zakomentuj dwie linie głównego pliku (są one opisane komentarzami w głównej stronie szablonu). Swoje synonimy wpisz do pliku o nazwie `synonimy.tex`.
7. Szablon korzysta z pakietu `cleveref` do bardziej elastycznego stosowania referencji wewnątrz dokumentu. Sprawdź możliwości tego pakietu: na przykład sprawdź co da skorzystanie z polecenia `\cref{. .}` zamiast standardowego `\ref{. .}` do odwoływania się do rysunków lub tabel.

Podziel swoją pracę na szereg mniejszych plików (na przykład: `Rozdzial01.tex`, `Rozdzial02.tex`, itd). Rozdziały te dołączaj do głównego pliku pracy za pomocą poleceń **include** lub **input** (include forsuje wstawienie polecenia nowej strony). Rozdziały umieszczaj w podkatalogu `chapters`.

2.1. Definicje użytkownika

Jeśli w pracy kilka razy posługujesz się tym samym fragmentem tekstu, to warto zdefiniować swoje własne polecenia LaTeX'a. Oto przykład wykorzystania definicji takich jak `\newcommand{\NN}{\mathbb{N}}` z głównego pliku tego szablonu. Oto przykład:

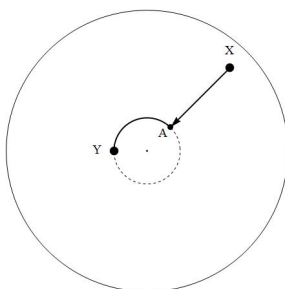
Symbolem $\mathbb{R}$ będziemy oznaczać zbiór liczb rzeczywistych, zaś symbolem $\mathbb{N}$ będziemy oznaczać zbiór liczb naturalnych. Zwróćmy uwagę na to, że liczbę 0 zaliczamy do zbioru liczb naturalnych, czyli $\mathbb{N} = \{0, 1, 2, \dots\}$.

Stosowanie takich definicji znacznie ujednolici tekst pracy oraz eliminuje szereg błędów (co jest chyba oczywiste dla każdego informatyka).

2.2. Ilustracje

Za pomocą polecenia `\includegraphics` można wstawiać obrazy do dokumentu LaTeX'a (potrzebna jest do tego celu, dołączona do szablonu, klasa `graphicx`). Obrazy umieszczaj w podkatalogu `figures`.

Dokładny opis parametrów tego polecenia możesz znaleźć na stronie https://latexref.xyz/_005cincludegraphics.html.



Bardziej elastyczny sposób polega na otoczeniu grafiki konstrukcją `picture` (patrz Rys. Rys. 2.1). Poniżej znajduje się Rys. 2.1.

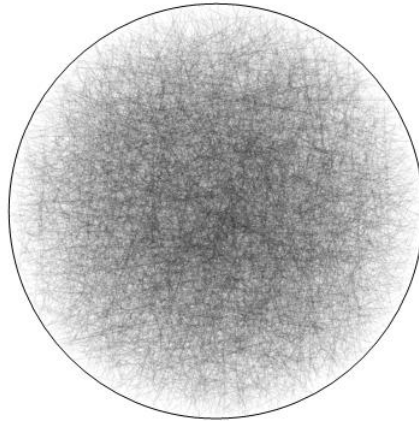
Szablon pracy dyplomowej wykorzystuje pakiet `cleveref`. Służy on do elastycznego operowania referencjami wewnątrz dokumentu. Na przykład, używając komendy `cref` (zamiast standardowej `\ref{fig:Lines}`), nie musisz samemu pisać skrótu Rys.

Jeśli odkomentujesz polecenie `\listoffigures` w głównym pliku, to rysunek ten znajdzie się na wygenerowanej liście rysunków.

2.3. Wzory matematyczne

Wzory matematyczne w obrębie paragrafu formatuje się za pomocą polecenia `$ \dots $`, np. $f(x) = \sin(x^2)$. Za pomocą polecenia `\[ \dots \]` formatujesz równania wycelowane bez numerowania, np.

$$\sin^2(\alpha) + \cos^2(\alpha) = 1 ,$$



Rysunek 2.1. Symulacje 10000 losowych odcinków w kółku jednostkowym. Rysunek został samodzielnie wygenerowany.

a za pomocą konstrukcji `\begin{equation} ... \end{equation}` formatujesz równania numerowane, np.

$$\exp(I \cdot t) = \cos(t) + I \cdot \sin(t) \quad (2.1)$$

Równań numerowanych używaj **tylko wtedy**, gdy będziesz się do nich odwoływać w dalszej części tekstu za pomocą polecenia `\ref{...}`.

Jeśli w pracy znajdują się jakieś twierdzenia, lematy, to skorzystaj z konstrukcji `theorem`, `lemma`, `corollary` oraz `proof`, zdefiniowanych w pakiecie `amsthm`, zaimportowanym do tego szablonu. Oto przykład:

Twierdzenie 2.1 (Euklides). *Istnieje nieskończenie wiele liczb pierwszych.*

Dowód. Załóżmy, że $p_1, \dots, p_n$ jest listą wszystkich liczb pierwszych. Niech

$$q = (p_1 \cdot \dots \cdot p_n) + 1.$$

Wtedy $q > 1$, istnieć więc musi liczba pierwsza p taka, że $p|q$

□

2.4. Cytowania

W tym szablonie korzystamy z biblioteki `biber` do formatowania cytatów. Jest to nowocześniejsza wersja standardowej (ale obecnie już nie rozwijanej) biblioteki `bibtex`. Oto przykład użycia:

Książka [2] jest poświęcona podstawowym algorytmom. Autor Knuth omawia w niej większość klasycznych algorytmów. Z punktu widzenia tej pracy ważny jest algorytm znajdujący się w Knuth [3, s. 249].

Prace [4], [5] zawierają zwięzłe wprowadzenia do problematyki *routingu geograficznego*.

Wpisy opisujące źródła bibliograficzne najlepiej jest przechowywać w oddzielnym pliku o formacie `bibtex`. Do szablonu dołączony jest przykładowy plik `bibliography.bib`.

Znajdują się w nim przykłady podstawowych typów tzw. wpisów. Porządnie sformatowane wpisy bibliograficzne możesz znaleźć, na przykład, na stronach [dblp](#) oraz w serwisie [Google Scholar](#).

Biber, używany z pakietem biblatex, zapewnia wiele predefiniowanych stylów formatowania cytatów i bibliografii, z typowymi stylami takimi jak `numeric`, `alphabetic`, `authoryear`, `authortitle` i `verbatim`, wraz z opcjami sortowania, takimi jak `nty` (name, title, year) lub `nyt` (name, year, title). Style te są wybierane za pomocą opcji `style=` w poleceniu

```
\usepackage[backend=biber, style=...]{biblatex}
```

w preambule LaTeX'a. Stosowany w pracy styl formatowania ustal wspólnie z opiekunem pracy. W pliku `main.tex` zaproponowano styl numeryczny bez opcji sortowania, co oznacza, że bibliografia będzie wyświetlana w kolejności pojawienia się cytowanej pozycji w tekście pracy.

Uwaga

Cytowania w LaTeX'u możesz samodzielnie obsługiwać bez pomocy pliku w formacie bibtex. Jednak samodzielne, poprawne sformatowanie bibliografii jest dość żmudne. Pamiętaj również o tym, że każda pozycja występująca w spisie literatury musi być cytowana w tekście pracy.

2.5. Tabele

Tabele wstawiasz w sposób standardowy. Po odkomentowaniu polecenia `\listoftables` z głównego pliku tego szablonu do pliku zostanie dołączona lista tabel. Do tej tabeli możesz się odwoływać za pomocą polecenia `\cref{tab:example}` (Tab. 2.1).

Tabela 2.1. Przykładowa tabela

Kategoria	Parametr	Wynik
Test A	Niskie obciążenie	15 ms
Test B	Wysokie obciążenie	120 ms

2.6. Kody procedur

Do tego szablonu dołączony został pakiet `listings`. Ułatwia on wstawianie kodów źródłowych do dokumentu. W szablonie, w pliku `ustawienia.tex`, są ustawione domyślne parametry wyświetlania dla kodów.

Oto przykład jego użycia:

```
1 /*
2  * Algorytm Euklidesa.
3  * Zwraca gcd(a, b).
4  */
5 int gcd(int a, int b)
6 {
```

```

7   while (b != 0) {
8       int r = a % b;
9       a = b;
10      b = r;
11  }
12  return a;
13 }

```

Listing 2.1. Algorytm Euklidesa w języku C

Oto inny przykład, kod implementacji algorytmu QuickSort w języku Haskell (listing 2.2) wykorzystujący, dla zwiększenia czytelności kodu, mechanizm *list comprehension*:

```

1  -- QuickSort: czas O(n^2); średni czas O(n log(n))
2  qs [] = []
3  qs (x:xs) = qs [t | t<- xs, t < x] ++
4              [x] ++
5              qs [t | t<- xs, t >= x]

```

Listing 2.2. Algorytm QuickSort w języku Haskell

Do wstawiania pseudokodów do pracy zalecamy korzystanie z pakietów `algorithm` oraz `algpseudocode` dołączonych do tego szablonu. Oto przykład pseudokodu zapisanego w języku tych pakietów (algorytm 1):

Algorytm 1 Filtrowanie

```

1: Initialize:  $S \leftarrow \emptyset$ 
2: for  $v \in V$  do
3:   if condition( $v$ ) then
4:      $S \leftarrow S \cup \{v\}$ 
5:   end if
6: end for
7: return  $S$ 

```

▷ Dodaj jeśli spełnia warunek

2.7. Akronimy

Szablon ten korzysta z pakietu `acro`. Plik `acronims.tex` jest przykładem pliku konfiguracyjnego. Oto przykład użycia:

Podczas budowania oprogramowania dużą wagę przykładaliśmy do przestrzegania dwóch zasad: zasady Don't repeat yourself (DRY) oraz zasady Keep It Simple, Stupid (KISS)

Za drugim razem, gdy użyjesz synonimów, pojawią się już tylko skrócone nazwy, na przykład:

Stosowanie zasady DRY polega unikaniu pisania kilkakrotnie podobnych fragmentów kodu. Stosowanie tej zasady znacznie ułatwia modyfikację kodów. Z kolei stosowanie zasady KISS zmusza programistę do systematycznego zastanawiania się nad jakością swojego kodu.

Jeśli nie masz potrzeby korzystania z akronimów, to zakomentuj dwie linie w pliku `main.tex` (linie 59 oraz 73).

Spis literatury

- [1] PWr WIT. *Wytyczne dot. SI na WIT*. 2025. URL: https://wit.pwr.edu.pl/wydzial/jakosc_ksztalcenia/wytyczne-dot-narzedzia-si-na-wit.
- [2] Donald E. Knuth. *The art of computer programming. Volume 1: Fundamental Algorithms*. 3rd. T. 1. The art of computer programming. Reading, Massachusetts: Addison-Wesley, 1997.
- [3] Donald E. Knuth. *The art of computer programming. Volume 2: Seminumerical Algorithms*. 3rd. T. 2. The art of computer programming. Reading, Massachusetts: Addison-Wesley, 1997.
- [4] Fabian Kuhn, Roger Wattenhofer i Aaron Zollinger. „An algorithmic approach to geographic routing in ad hoc and sensor networks”. W: *IEEE/ACM Trans. Netw.* 16.1 (2008), s. 51–62.
- [5] L. Popa i in. „Balancing Traffic Load in Wireless Networks with Curveball Routing”. W: *Mobihoc'07*. 2007.