



# Skrypt z Algorytmów i struktur danych

*Zbiór mniej lub bardziej ciekawych algorytmów i struktur danych, jakie bywały omawiane na wykładzie (albo nie).*

PRACA ZBIOROWA POD REDAKCJĄ  
KRZYSZTOFA PIECUCHA

Korzystać na własną odpowiedzialność.



# Spis treści

<b>1</b>	<b>Podstawy</b>	<b>5</b>
1.1	Złożoność obliczeniowa . . . . .	6
1.2	Model obliczeń . . . . .	7
<b>2</b>	<b>Struktury danych</b>	<b>9</b>
2.1	Kopce binarne . . . . .	10
<b>3</b>	<b>Algorytmy</b>	<b>13</b>
3.1	Algorytm rosyjskich wieśniaków . . . . .	14
3.2	Algorytm macierzowy wyznaczania liczb Fibonacciego . . . . .	15
3.3	Sortowanie topologiczne . . . . .	16
3.4	Algorytmy sortowania . . . . .	17
3.5	Minimalne drzewa rozpinające . . . . .	18
3.5.1	Cut Property i Circle Property . . . . .	18
3.5.2	Algorytm Prima . . . . .	18
3.5.3	Algorytm Kruskala . . . . .	18
3.5.4	Algorytm Borůvky . . . . .	18
3.6	Algorytm Dijkstry . . . . .	19
3.7	Algorytm szeregowania . . . . .	20
3.8	Programowanie dynamiczne na drzewach . . . . .	21
<b>Dodatek A Porównanie programów przedmiotu AiSD na różnych uczelniach</b>		<b>23</b>



# Rozdział 1

## Podstawy

## 1.1 Złożoność obliczeniowa

Todo, todo, todo...

## 1.2 Model obliczeń

Todo, todo, todo...





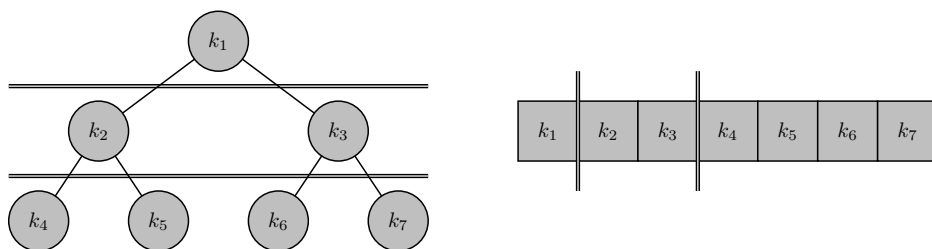
## Rozdział 2

# Struktury danych

## 2.1 Kopce binarne

Kopiec binarny to struktura danych, która reprezentowana jest jako prawie pełne drzewo binarne<sup>1</sup> i na której zachowana jest własność kopca. Kopiec przechowuje klucze, które tworzą ciąg uporządkowany. W przypadku kopca typu *min* ścieżka prowadząca od dowolnego liścia do korzenia tworzy ciąg malejący.

Kopce można w prosty sposób reprezentować w tablicy jednowymiarowej – kolejne poziomy drzewa zapisywane są po sobie.



Rysunek 2.1: Reprezentacja kolejnych warstw kopca w tablicy jednowymiarowej.

Warto zauważyć, że tak reprezentowane drzewo pozwala na łatwy dostęp do powiązanych węzłów. Synami węzła o indeksie  $i$  są węzły  $2i$  oraz  $2i + 1$ , natomiast jego ojcem jest  $\lfloor \frac{i}{2} \rfloor$ .

Kopiec powinien udostępniać trzy podstawowe funkcje: `zamien_element`, która podmienia wartość w konkretnym węźle kopca, `przesun_w_gore` oraz `przesun_w_dol`, które zamieniają odpowiednie elementy pilnując przy tym, aby własność kopca została zachowana.

---

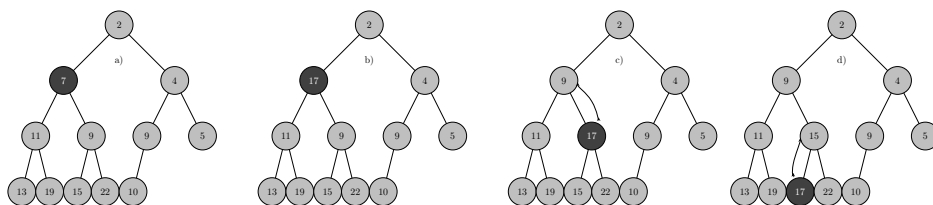
**Algorithm 1:** Implementacja funkcji `zamien_element`

---

```
if  $k[i] < v$  then
     $k[i] = v$ ;
    przesun_w_dol(k, i);
end
else
     $k[i] = v$ ;
    przesun_w_gore(k, i);
end
```

---

<sup>1</sup>To znaczy wypełniony na wszystkich poziomach (poza, być może, ostatnim).



Rysunek 2.2: Przykład działania funkcji `zamien_element`. a) Oryginalny kopiec. b) Zmiana wartości w wyróżnionym węźle. c) Ponieważ nowa wartość jest większa od wartości swoich dzieci, należy wykonać wywołanie funkcji `przesn_w_dol`. d) Po zmianie własność kopca nie jest zachowana, dlatego należy ponownie wywołać funkcję `przesn_w_dol`. To przywraca kopcowi jego własność.

;



## Rozdział 3

# Algorytmy

### 3.1 Algorytm rosyjskich wieśniaków

Todo, todo, todo...

---

**Algorithm 2:** Algorytm rosyjskich wieśniaków

---

**Input:**  $a, b$  - liczby naturalne

**Output:**  $wynik = a \cdot b$

$a' \leftarrow a$

$b' \leftarrow b$

$wynik \leftarrow 0$

**while**  $a' > 0$  **do**

**if**  $a' \bmod 2 = 1$  **then**

$wynik \leftarrow wynik + b'$

**end**

$a' \leftarrow a'/2$

$b' \leftarrow b' \cdot 2$

**end**

---

## 3.2 Algorytm macierzowy wyznaczania liczb Fibonacciego

Todo, todo, todo...

### 3.3 Sortowanie topologiczne



Rysunek 3.1: Przykładowy graf z ubraniami dla bramkarza hokejowego. Krawędź między wierzchołkami  $a$  oraz  $b$  istnieje wtedy i tylko wtedy, gdy gracz musi ubrać  $a$  zanim ubierze  $b$ . Pytanie o to w jakiej kolejności bramkarz powinien się ubierać, jest pytaniem o posortowanie topologiczne tego grafu.



## 3.4 Algorytmy sortowania

Todo, todo, todo...

## 3.5 Minimalne drzewa rozpinające

Todo, todo, todo...

### 3.5.1 Cut Property i Circle Property

### 3.5.2 Algorytm Prima

### 3.5.3 Algorytm Kruskala

### 3.5.4 Algorytm Borůvky

## 3.6 Algorytm Dijkstry

Todo, todo, todo...

## 3.7 Algorytm szeregowania

Todo, todo, todo...

## 3.8 Programowanie dynamiczne na drzewach

Todo, todo, todo...





## Dodatek A

# Porównanie programów przedmiotu AiSD na różnych uczelniach

	UWr	UW	UJ	MIT	Oxford
Stosy, kolejki, listy		✓			
Dziel i zwyciężaj	✓				
Programowanie Dynamiczne	✓	✓	✓	✓	
Metoda Zachłanna	✓	✓	✓		
Koszt zamortyzowany	✓	✓			✓
NP-zupełność	✓	✓		✓	
PRAM / NC	✓				
Sortowanie	✓	✓			
Selekcja	✓	✓			
Słowniki	✓	✓	✓		✓
Kolejki priorytetowe	✓	✓			
Hashowanie	✓	✓			
Zbiory rozłączne	✓				
Algorytmy grafowe	✓	✓	✓	✓	✓
Algorytmy tekstowe	✓	✓			
Geometria obliczeniowa	✓				
FFT	✓				✓
Algorytm Karatsuby	✓			✓	
Metoda Newtona				✓	
Algorytmy randomizowane	✓				✓
Programowanie liniowe					✓
Algorytmy aproksymacyjne	✓				✓
Sieci komparatorów	✓				
Obwody logiczne	✓				