

# Ile potrzeba matematyki, żeby zrozumieć prosty algorytm ?

Marek Klonowski

Marek.Klonowski@pwr.edu.pl

Katedra Informatyki  
Wydział Podstawowych Problemów Techniki  
Politechnika Wroclawska

2 grudnia 2016 r.



## Zainteresowania naukowe

- informatyka teoretyczna: bezpieczeństwo informacji (podpisy cyfrowe, anonimowa komunikacja), algorytmy rozproszone
- matematyka: probabilistyka

## Dydaktyka

- matematyka: rachunek prawdopodobieństwa, kombinatoryka analityczna
- informatyka: algorytmy zrandomizowane, algorytmy rozproszone, kryptografia
- doświadczenie w pracy z uczniami liceów



## Zainteresowania naukowe

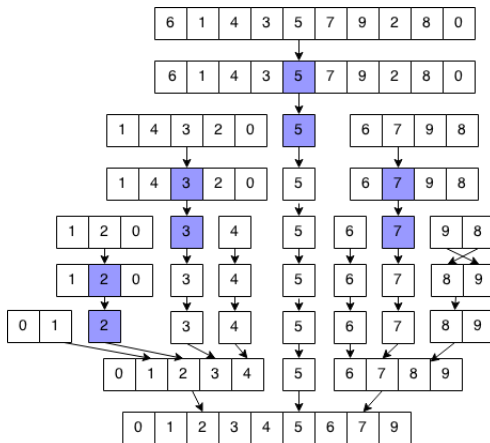
- informatyka teoretyczna: bezpieczeństwo informacji (podpisy cyfrowe, anonimowa komunikacja), algorytmy rozproszone
- matematyka: probabilistyka

## Dydaktyka

- matematyka: rachunek prawdopodobieństwa, kombinatoryka analityczna
- informatyka: algorytmy zrandomizowane, algorytmy rozproszone, kryptografia
- doświadczenie w pracy z uczniami liceów



# Sortowanie szybkie (Quick Sort) - QS



Rysunek: QS



---

## Algorithm 1 QS pseudocode

---

```
1:  $QS(Tab, left, right)$ 
2: if  $left < right$  then
3:   select a  $PivotIndex$ 
4:    $PivotNewIndex := partition(Tab, left, right, PivotIndex)$ 
5:    $QS(Tab, left, pivotNewIndex - 1)$ 
6:    $QS(Tab, pivotNewIndex + 1, right)$ 
7: end if
```

---



---

## Algorithm 2 RQS pseudocode

---

```
1: RQS(Tab, left, right)
2: if left < right then
3:   select a random PivotIndex
4:   PivotNewIndex := partition(Tab, left, right, PivotIndex)
5:   RQS(Tab, left, pivotNewIndex - 1)
6:   RQS(Tab, pivotNewIndex + 1, right)
7: end if
```

---



## Pojęcia wstępne

- Wejście (Input) - to **permutacja**
- *a random* - po polsku: losowy z jednostajnym **p-stwem**
- *RQS* wywołuje *RQS* - abstrakcyjne i nieprzyjemnie (**rekursja**)

## Co chcemy analizować ?

- Liczba porównań
- $Q_n$  - liczba porównań dla wejścia długości  $n$



## Pojęcia wstępne

- Wejście (Input) - to **permutacja**
- *a random* - po polsku: losowy z jednostajnym **p-stwem**
- *RQS* wywołuje *RQS* - abstrakcyjne i nieprzyjemnie (**rekursja**)

## Co chcemy analizować ?

- Liczba porównań
- $Q_n$  - liczba porównań dla wejścia długości  $n$





## Pojęcia wstępne

- Wejście (Input) - to **permutacja**
- *a random* - po polsku: losowy z jednostajnym **p-stwem**
- *RQS* wywołuje *RQS* - abstrakcyjne i nieprzyjemnie (**rekursja**)

## Co chcemy analizować ?

- Liczba porównań
- $Q_n$  - liczba porównań dla wejścia długości  $n$



## Pojęcia wstępne

- Wejście (Input) - to **permutacja**
- *a random* - po polsku: losowy z jednostajnym **p-stwem**
- *RQS* wywołuje *RQS* - abstrakcyjne i nieprzyjemnie (**rekursja**)

## Co chcemy analizować ?

- Liczba porównań
- $Q_n$  - oczekiwana liczba ~~liczba porównań~~ dla wejścia długości  $n$  (**ciąg**)



## Fundamentalne spostrzeżenie

$$Q_n = (n-1) + \frac{1}{n} \sum_{k=0}^{n-1} (Q_k + Q_{n-k-1})$$

$$Q_0 = Q_1 = 0$$

Co trzeba umieć ?

- p-stwo warunkowe
- p-stwo całkowite



## Fundamentalne spostrzeżenie

$$Q_n = (n-1) + \frac{1}{n} \sum_{k=0}^{n-1} (Q_k + Q_{n-k-1})$$

$$Q_0 = Q_1 = 0$$

## Co trzeba umieć ?

- p-stwo warunkowe
- p-stwo całkowite



## Przekształcenia

$$① \quad Q_n = (n-1) + \frac{1}{n} \sum_{k=0}^{n-1} (Q_k + Q_{n-k-1})$$

$$② \quad Q_n = (n-1) + \frac{2}{n} \sum_{k=0}^{n-1} Q_k$$

$$③ \quad nQ_n = n(n-1) + 2 \sum_{k=0}^{n-1} (Q_k + Q_{n-k-1})$$



## Przekształcenia

$$1 \quad Q_n = (n-1) + \frac{1}{n} \sum_{k=0}^{n-1} (Q_k + Q_{n-k-1})$$

$$2 \quad Q_n = (n-1) + \frac{2}{n} \sum_{k=0}^{n-1} Q_k$$

$$3 \quad nQ_n = n(n-1) + 2 \sum_{k=0}^{n-1} (Q_k + Q_{n-k-1})$$



## Przekształcenia

$$\textcircled{1} \quad Q_n = (n-1) + \frac{1}{n} \sum_{k=0}^{n-1} (Q_k + Q_{n-k-1})$$

$$\textcircled{2} \quad Q_n = (n-1) + \frac{2}{n} \sum_{k=0}^{n-1} Q_k$$

$$\textcircled{3} \quad nQ_n = n(n-1) + 2 \sum_{k=0}^{n-1} (Q_k + Q_{n-k-1})$$



## Przekształcenia

$$1 \quad nQ_n = 2(n-1) + (n+1)Q_{n-1}$$

$$2 \quad \frac{Q_n}{n+1} = 2 \frac{n-1}{n(n+1)} + \frac{Q_{n-1}}{n}$$

$$3 \quad F_n := \frac{Q_n}{n+1}$$

$$4 \quad F_n = 2 \frac{n-1}{n(n+1)} + F_{n-1} = 2 \sum_{k=1}^n \frac{k-1}{k(k+1)}$$





## Przekształcenia

$$1 \quad nQ_n = 2(n-1) + (n+1)Q_{n-1}$$

$$2 \quad \frac{Q_n}{n+1} = 2\frac{n-1}{n(n+1)} + \frac{Q_{n-1}}{n}$$

$$3 \quad F_n := \frac{Q_n}{n+1}$$

$$4 \quad F_n = 2\frac{n-1}{n(n+1)} + F_{n-1} = 2 \sum_{k=1}^n \frac{k-1}{k(k+1)}$$



## Przekształcenia

$$1 \quad nQ_n = 2(n-1) + (n+1)Q_{n-1}$$

$$2 \quad \frac{Q_n}{n+1} = 2\frac{n-1}{n(n+1)} + \frac{Q_{n-1}}{n}$$

$$3 \quad F_n := \frac{Q_n}{n+1}$$

$$4 \quad F_n = 2\frac{n-1}{n(n+1)} + F_{n-1} = 2 \sum_{k=1}^n \frac{k-1}{k(k+1)}$$



## Przekształcenia

$$① \quad nQ_n = 2(n-1) + (n+1)Q_{n-1}$$

$$② \quad \frac{Q_n}{n+1} = 2\frac{n-1}{n(n+1)} + \frac{Q_{n-1}}{n}$$

$$③ \quad F_n := \frac{Q_n}{n+1}$$

$$④ \quad F_n = 2\frac{n-1}{n(n+1)} + F_{n-1} = 2\sum_{k=1}^n \frac{k-1}{k(k+1)}$$



## Przekształcenia

$$1 \quad F_n = 2 \frac{n-1}{n(n+1)} + F_{n-1} = 2 \sum_{k=1}^n \frac{k-1}{k(k+1)}$$

$$2 \quad \text{Trik: } \frac{k-1}{k(k+1)} = \frac{2}{1+k} - \frac{1}{k}$$

$$3 \quad F_n = 4 \sum_{k=1}^n \frac{1}{1+k} - 2 \sum_{k=1}^n \frac{1}{k}$$

$$4 \quad H_n := \sum_{k=1}^n \frac{1}{k}$$



## Przekształcenia

$$1 \quad F_n = 2 \frac{n-1}{n(n+1)} + F_{n-1} = 2 \sum_{k=1}^n \frac{k-1}{k(k+1)}$$

$$2 \quad \text{Trik: } \frac{k-1}{k(k+1)} = \frac{2}{1+k} - \frac{1}{k}$$

$$3 \quad F_n = 4 \sum_{k=1}^n \frac{1}{1+k} - 2 \sum_{k=1}^n \frac{1}{k}$$

$$4 \quad H_n := \sum_{k=1}^n \frac{1}{k}$$



## Przekształcenia

$$1 \quad F_n = 2 \frac{n-1}{n(n+1)} + F_{n-1} = 2 \sum_{k=1}^n \frac{k-1}{k(k+1)}$$

$$2 \quad \text{Trik: } \frac{k-1}{k(k+1)} = \frac{2}{1+k} - \frac{1}{k}$$

$$3 \quad F_n = 4 \sum_{k=1}^n \frac{1}{1+k} - 2 \sum_{k=1}^n \frac{1}{k}$$

$$4 \quad H_n := \sum_{k=1}^n \frac{1}{k}$$



## Przekształcenia

$$1 \quad F_n = 2 \frac{n-1}{n(n+1)} + F_{n-1} = 2 \sum_{k=1}^n \frac{k-1}{k(k+1)}$$

$$2 \quad \text{Trik: } \frac{k-1}{k(k+1)} = \frac{2}{1+k} - \frac{1}{k}$$

$$3 \quad F_n = 4 \sum_{k=1}^n \frac{1}{1+k} - 2 \sum_{k=1}^n \frac{1}{k}$$

$$4 \quad H_n := \sum_{k=1}^n \frac{1}{k}$$



## Przekształcenia

1  $Q_n = (n + 1)(4H_{n+1} - 2H_n - 4)$

2  $H_n = \ln n + \gamma + \frac{1}{2n} + O\left(\frac{1}{n^2}\right)$  AM + logarytmy

3  $Q_n = 2n(\ln n + \gamma - 2) + 2 \ln n + 2\gamma + 1 + O\left(\frac{1}{n}\right)$





## Przekształcenia

1  $Q_n = (n + 1)(4H_{n+1} - 2H_n - 4)$

2  $H_n = \ln n + \gamma + \frac{1}{2n} + O\left(\frac{1}{n^2}\right)$  **AM + logarytmy**

3  $Q_n = 2n(\ln n + \gamma - 2) + 2 \ln n + 2\gamma + 1 + O\left(\frac{1}{n}\right)$



## Przekształcenia

$$① \quad Q_n = (n+1)(4H_{n+1} - 2H_n - 4)$$

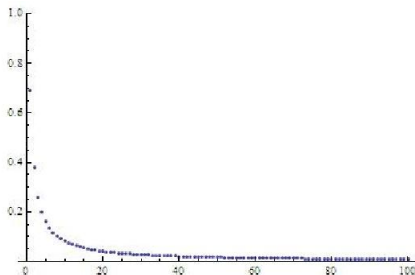
$$② \quad H_n = \ln n + \gamma + \frac{1}{2n} + O\left(\frac{1}{n^2}\right) \text{ AM + logarytmy}$$

$$③ \quad Q_n = 2n(\ln n + \gamma - 2) + 2 \ln n + 2\gamma + 1 + O\left(\frac{1}{n}\right)$$



## Końcowy wynik

$$Q_n = 2n(\ln n + \gamma - 2) + 2\ln n + 2\gamma + 1 + O\left(\frac{1}{n}\right)$$



Rysunek: Błąd oszacowania  $Q_n$



## Czy warto ?

- Anonimowy głos: *Nie warto się uczyć algorytmów. **Wszystkie** algorytmy są już zaimplementowane.*
- `randomizedQuickSort.java`
- Potrzeba znać aby dostać **dobrą** pracę w IT
- Obecnie rynek pracy w IT jest bardzo dobry. Przyszłość: Chiny, Indie, Ukraina, Białoruś



## Czy warto ?

- Anonimowy głos: *Nie warto się uczyć algorytmów.*  
**Wszystkie** algorytmy są już zaimplementowane.
- `randomizedQuickSort.java`
- Potrzeba znać aby dostać **dobrą** pracę w IT
- Obecnie rynek pracy w IT jest bardzo dobry. Przyszłość: Chiny, Indie, Ukraina, Białoruś



## Czy warto ?

- Anonimowy głos: *Nie warto się uczyć algorytmów. **Wszystkie** algorytmy są już zaimplementowane.*
- `randomizedQuickSort.java`
- Potrzeba znać aby dostać **dobrą** pracę w IT
- Obecnie rynek pracy w IT jest bardzo dobry. Przyszłość: Chiny, Indie, Ukraina, Białoruś



## Czy warto ?

- Anonimowy głos: *Nie warto się uczyć algorytmów.*  
***Wszystkie*** algorytmy są już zaimplementowane.
- `randomizedQuickSort.java`
- Potrzeba znać aby dostać **dobrą** pracę w IT
- Obecnie rynek pracy w IT jest bardzo dobry. Przyszłość: Chiny, Indie, Ukraina, Białoruś



## Czy warto ?

- Anonimowy głos: *Nie warto się uczyć algorytmów. **Wszystkie** algorytmy są już zaimplementowane.*
- `randomizedQuickSort.java`
- Potrzeba znać aby dostać **dobrą** pracę w IT
- Obecnie rynek pracy w IT jest bardzo dobry. Przyszłość: Chiny, Indie, Ukraina, Białoruś





## Temat przewodni:

*O potrzebie innowacyjnego kształcenia uczniów i studentów*

## Co MSZ jest istotne

- Dobre rozumienie pojęć, nie schematy, abstrakcja
- Synergia: matematyka  $\sim$  informatyka



## Temat przewodni:

*O potrzebie innowacyjnego kształcenia uczniów i studentów*

## Co MSZ jest istotne

- Dobre rozumienie pojęć, nie schematy, abstrakcja
- Synergia: matematyka  $\sim$  informatyka



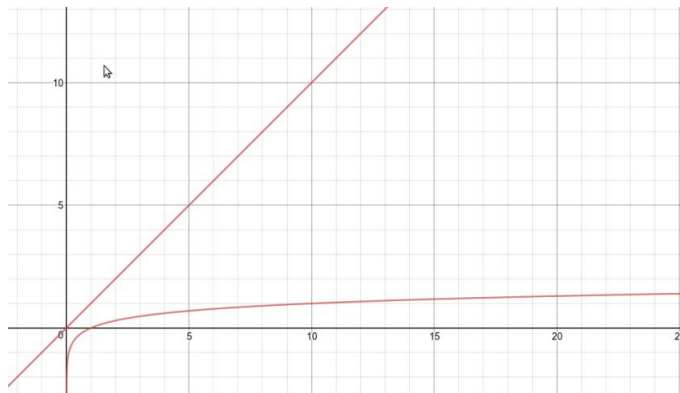
## Temat przewodni:

*O potrzebie innowacyjnego kształcenia uczniów i studentów*

## Co MSZ jest istotne

- Dobre rozumienie pojęć, nie schematy, abstrakcja
- Synergia: matematyka  $\sim$  informatyka





Rysunek:  $x$  i  $\log n$



## Kontakt:

- email: [Marek.Klonowski@pwr.edu.pl](mailto:Marek.Klonowski@pwr.edu.pl)
- WWW: <http://cs.pwr.edu.pl/klonowski/>

# Dziękuję

