

# Bloom Filter

Florian Eberhard Schierz  
TU Bergakademie Freiberg  
Freiberg, Germany

florian-eberhard.schierz@student.tu-freiberg.de

## ZUSAMMENFASSUNG

Bloomfilter sind probabilistische Datenstrukturen, mit denen platzsparend große Mengen an Elementen gespeichert und auf Mitgliedschaft geprüft werden können. To be continued...

## KEYWORDS

Datenstrukturen, Bloomfilter, probabilistisch, Hash

### ACM Reference Format:

Florian Eberhard Schierz. 2022. Bloom Filter. In *Proceedings of Seminar on Data Structures in C and Ubiquitous Computing (UbiSys Seminar '22)*. ACM, New York, NY, USA, 2 pages.

## 1 EINLEITUNG

Thema -> Bloomfilter und seine Eigenschaften, Varianten und Einsatzgebiete

Einordnung in wissenschaftlichen Kontext und Stand der Forschung  
-> Einführung 1970 durch Burton Bloom, seitdem häufig in Praxis und Literatur thematisiert, dadurch unzählige Varianten und Implementierungen

Ziel der Arbeit -> Übersicht über BF allgemein, welche Varianten es mit welchen Eigenschaften gibt und wo man BF einsetzt, samt eigener Implementierung

Die verwendete Untersuchungsmethodik -> Literaturrecherche und Implementierung

## 2 DER STANDARD BLOOMFILTER

von Burton Bloom 1970 entwickelt, probabilistisch, platzsparend Elemente merken, ohne wirklich zu speichern, somit nur auf Mitgliedschaft prüfen, Array in dem 0 und 1 setzt, Position mit Hash Fkt ermittelt, rückwärts so auf Mitgliedschaft kontrollieren, dabei können Fehler entstehen aufgrund von nicht bijektivität der Hash Fkt

### 2.1 Parameter des Standard Bloomfilters

n Elemente einzuspeichern, k Hash Fkt, m Arraygröße, FPP

Formeln wie diese zsm hängen

Siehe Implementierung (darauf verweisen)

### 2.2 Operationen und Komplexität

Einfügen und Überprüfen beide Abhängig von Hash Anzahl k  
Implementation verweisen, Zeitmessung

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

UbiSys Seminar '22, Summer term 2022, TU Freiberg, DE

© 2022 Copyright held by the owner/author(s).

## 3 VARIANTEN DES BLOOMFILTERS

kurze Beschreibung der bekanntesten Varianten:

Hier muss noch aussortiert werden, welche am Ende wirklich beschreiben werden, diese Varianten habe ich aber alle gefunden  
mund Material dazu herausgesucht

### 3.1 Counting Blomfilter

### 3.2 Variable Increment Blomfilter

### 3.3 Compressed Blomfilter

### 3.4 Scalable Blomfilter

### 3.5 Generalized Blomfilter

### 3.6 Bloomier Blomfilter

### 3.7 Stable Blomfilter

### 3.8 Weighted Blomfilter

### 3.9 Deletable Blomfilter

### 3.10 Spectral Blomfilter

### 3.11 Robust Blomfilter

### 3.12 Floomfilter

evtl. eigene Variante je nachdem wie die Entwicklung läuft und ob es präsentierbar ist -> Verweis auf Implementation

## 4 VERGLEICH ZU ANDEREN STRUKTUREN

Auf vergleichbare eventuell bessere Strukturen eingehen, zB Cuckoo Filter

## 5 ANWENDUNGEN

Anwendungsmöglichkeiten beschreiben

### 5.1 Netzerkanwendungen

DDOS Preventing, Routing, Web caches

### 5.2 Sicherheitsanwendungen

Intrusion Detection, Encrypted Search

### 5.3 Weitere Anwendungen

rechtschreibsysteme, Longest Prefix Match

## 6 ZUSAMMENFASSUNG

Vielseitige Varianten, die Anwendung in vielen Bereichen ermöglichen und je nachdem an spezielle Anforderungen angepasst werden können

Sehr einfach zunutzen und daher dank Effizienz oft ausreichend und erste Wahl obwohl bessere Umsetzungen gibt

Weitere Forschungsmöglichkeiten sind weitere Variationen zu finden, die noch platzsparender und schneller sind um so weiter zu optimieren Beispielreferenz [1]

## LITERATUR

- [1] Mauro Conti, Roberto Di Pietro, Luigi V. Mancini, and Alessandro Mei. 2009. (old) Distributed data source verification in wireless sensor networks. *Inf. Fusion* 10, 4 (2009), 342–353. <https://doi.org/10.1016/j.inffus.2009.01.002>