# Einleitung

* Wenn BF sage, denken manche bestimmt an so etwas (klick)
* Aber nein ich habe keine Endung verschluckt
* Erzähle euch heute wirklich etwas über BF aber die werden so hier geschrieben

# Gliederung

* Vorlesen

# Prinzipielle Idee

* Was ist BF?
* Erfunden 1970 von Burton Bloom
* Überprüfung ob Elemente in bestimmter vorher eingespeicherter Menge enthalten oder eben nicht
* Dabei Elemente selbst bzw der Inhalt aber nicht gespeichert sondern quasi merkt nur dass schonmal gesehen
* Ähnlich wie Fingerabdruckdatenbank 🡪 weiß dass den Fingerabdruck schon gesehen
* Dadurch ist sehr platzsparend und effizient
* Haken an der Sache
* Probabilistisch = Hat mit Wahrscheinlichkeit zu tun
* Bedeutet dass Aussagen nicht immer verlässlich sondern nur zu bestimmter WK
* Dabei unterscheiden: Wenn nein sagt, stimmt sicher
* Bei „ja“ nur zu bestimmter WK
* So entsteht Falsch Positiv Rate (FPP)
* Prinzip auf Grafik erkennbar

# Anwendungen

* Was kann man mit diesem Prinzip nun anfangen?
* Wo helfen einem Aussagen die falsch sein können?
* Klingt vllt erstmal absurd aber gibt sehr viele Anwendungen
* Allgemein kann sagen: Wo schnell Mitgliedschaft eines Elements aus großer Menge geprüft werden soll und wo dabei kleiner Fehlerenateil nicht verheerend ist
* Im speziellen finden dafür oft Anwendung in NW gefunden
* Routing
  + Anfrage für Ressource und soll schnellstmöglich zu Server der sie hat
  + Wo geht lang?
  + Speicher für alle Nachbarn/Links BF mit Ressourcen die dort gespeichert
  + Somit schnell prüfen
  + Wenn kein direkter Nachbar hat, geht immer Ebene tiefer wo Dokumente der Nachbarn der Nachbar sind usw. bis gefunden und dann schickt weiter oder wenn nicht findet, gibt an deterministischen Algorithmus weiter, der länger braucht
* Loop Prevention
  + BF in Header der Pakete hinzu
  + Jeder Node hat Bloom Mask mit 1 und 0 drin (äquivalent zu seinen Hash Werten)
  + ORed das mit rumgeschickten BF
  + Wenn ändert, war hier noch nicht, wenn nicht ändert liegt Loop nahe
  + Oder FP 🡪 schlecht
* Trace Route
  + BF in Paket mit drin
  + Jeder Router trägt dort Hash Wert seiner IP vom Output Interface ein
  + Kaum Zeitaufwand, da vorher berechnet werden kann und dann nur OR
  + Rückverfolgung über Überprüfung der Nachbar IP Adressen
  + Findet so immer Vorgänger 🡪 rekursiv zurückwandern
* DDoS
  + Einmal darüber und dann IP ignorieren
  + Oder zum White- bzw. Blacklisten von IP Adressen
* Web Cache
  + Speichert URLs in BF ein die gecached hat
  + Somit schnelle Überprüfung möglich
  + Je nachdem anfragen oder schicken
* Weitere Anwendungen gehen in Richtung Sicherheit/Security
* IDS
  + Bekannte gefährliche Strings von Viren (Signatures genannt)
  + Diese in verschiedenen Längen gestückelt in verschiedenen BF gespeichert
  + Eintreffende Daten werden dann ebenso zerstückelt und jeweils auf Membership in einem der BF geprüft
  + Wenn enthalten markiert als potentiell gefährlich
  + Folgt Überprüfung mit deterministischem Verfahren da ja auch FP sein könnte
* Datenbanken
  + Differential Files werden in DB verwendet um Änderungen abzufangen und dann gesammelt durchzuführen
  + BF dabei verwendet um vorherzusagen ob Anfrage auf DF oder in DB geht, weil das eben noch nicht geändert wurde
  + Zeit sparen, weil Suche einschränkt
* Weitere praktische und alltägliche Anwendungen
* Rechtschreibprüfung
  + Alle Wörter einer Sprache einspeichern in BF (deutsch zB 500 000)
  + Prüft dann geschriebenes Wort auf Membership
  + Gibt keine Vorschläge zur Korrektur aber guter erster Indikator
  + Ähnlich wie iDS
* LPM
  + Wieder ähnlich IDS für Prefixe genutzt
  + Zerteilt IP Adresse in bitweise Stücke
  + Jeweils BF für die eingespeicherten Prefixe mit bestimmten Längen
  + So wieder schnell überprüfen ob zB in BF mit Prefixlänge 6 Match und 7 nicht mehr, dann weiß dass da ist
* Ähnlichkeitsanalysen bei Websuchen
  + Teilt Seiten in Chunks
  + Diese in BF gespeichert
  + Vergleicht BF mit AND
  + Wenn viele 1 übrig bleiben ist Seite ähnlich
  + Somit Suchergebnisse verbessern oder zumindest gruppieren