Søketek-compression!!

2022-09-28

Compression,
Encoding,
Ranked retrival,
Tips til oblig B,
Introdusere oblig C,
Et knippe eksamensoppgaver,
Oblighjelp.

+ Kanskje se på oblig A-repl

Compression

- Bruk mindre plass på å lagre noe
- Lossy/lossless compression
- Utnytt domenkunnskap
- E.g. kun lagre diffen mellom postings

Encoding

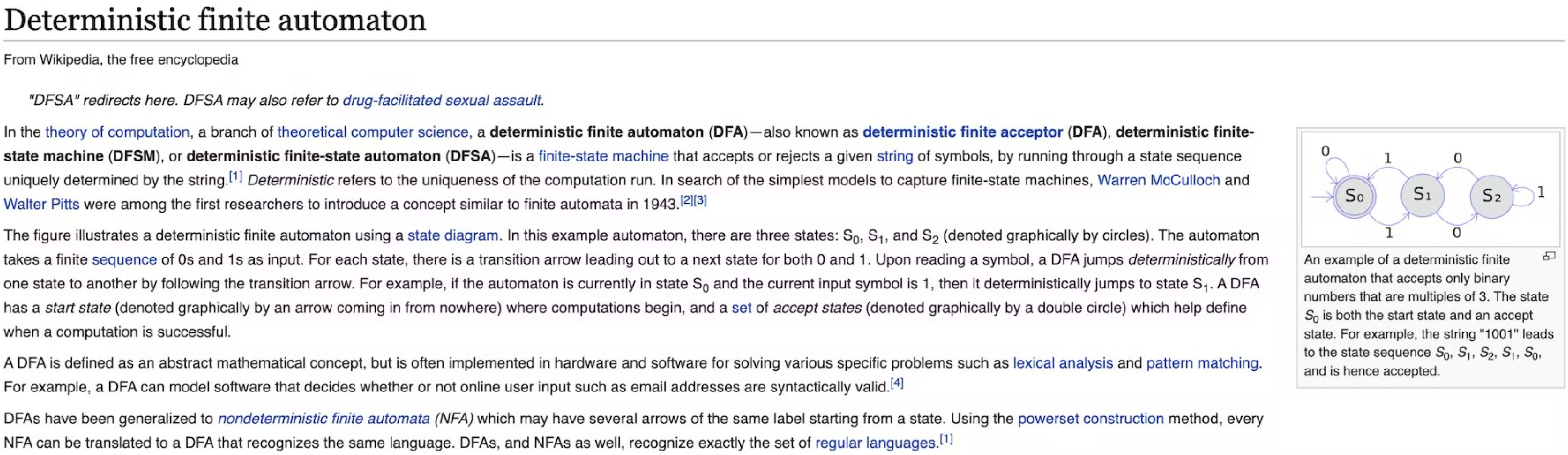
- VB -> Variable byte
- Gamma enoding -> Variable bit
- (Elias [gamma, delta] codes)

Compression buzzwords you should recognize

1. Simple9
2. PFOR-DELTA
3. Unary codes
4. Rice- & Golomb-coding

Ranked retrival

- "Hvilket resultat matcher queryen *best*?"
- Sorter resultatene før man returnerer dem
- TF, TF-IDF, cosine similarity
- Språkteknologiske heuristikker
- Se prekode `sieve.py`



Deterministic finite automaton

From Wikipedia, the free encyclopedia

"DFSA" redirects here. DFSA may also refer to drug-facilitated sexual assault.

In the theory of computation, a branch of theoretical computer science, a **deterministic finite automaton** (**DFA**)—also known as **deterministic finite acceptor** (**DFA**), **deterministic finite-state machine** (**DFSM**), or **deterministic finite-state automaton** (**DFSA**)—is a finite-state machine that accepts or rejects a given string of symbols, by running through a state sequence uniquely determined by the string.[1] *Deterministic* refers to the uniqueness of the computation run. In search of the simplest models to capture finite-state machines, Warren McCulloch and Walter Pitts were among the first researchers to introduce a concept similar to finite automata in 1943.[2][3]

The figure illustrates a deterministic finite automaton using a state diagram. In this example automaton, there are three states: S_0, S_1, and S_2 (denoted graphically by circles). The automaton takes a finite sequence of 0s and 1s as input. For each state, there is a transition arrow leading out to a next state for both 0 and 1. Upon reading a symbol, a DFA jumps *deterministically* from one state to another by following the transition arrow. For example, if the automaton is currently in state S_0 and the current input symbol is 1, then it deterministically jumps to state S_1. A DFA has a *start state* (denoted graphically by an arrow coming in from nowhere) where computations begin, and a set of *accept states* (denoted graphically by a double circle) which help define when a computation is successful.

A DFA is defined as an abstract mathematical concept, but is often implemented in hardware and software for solving various specific problems such as lexical analysis and pattern matching. For example, a DFA can model software that decides whether or not online user input such as email addresses are syntactically valid.[4]

DFAs have been generalized to *nondeterministic finite automata (NFA)* which may have several arrows of the same label starting from a state. Using the powerset construction method, every NFA can be translated to a DFA that recognizes the same language. DFAs, and NFAs as well, recognize exactly the set of regular languages.[1]

An example of a deterministic finite automaton that accepts only binary numbers that are multiples of 3. The state S_0 is both the start state and an accept state. For example, the string "1001" leads to the state sequence S_0, S_1, S_2, S_1, S_0, and is hence accepted.

Oblig B Trie-walk-algorithm. Stringfinder.scan() := DFA

Relevante papers for Tries

1. [A. V. Aho, M. J. Corasick 1975], "Efficient String Matching: An Aid to Bibliographic Search".
2. [H. Shang, T. H. Merrret 1995], "Tries for Approximate String Matching"
3. ([Germann et al 2009] "Tightly Packed Tries", mindre relevant)

Oblig C

- WAND-algoritmen ("Weak AND") (og en bra pun)
- "Gi meg alle dokumentene som inneholder minst m av disse n termene"*
- *'Lett' å brute-force, vi må gjøre det på en spesiell (effektiv) måte.
- *IKKE* iterer over korpuset i oblig C (viktig, se DaaT vs TaaT))
- Kun én prosedyre i 1 fil (simplesearchengine.py)

Random overraskende relevant PDF:

https://resources.mpi-inf.mpg.de/departments/d5/teaching/ws13_14/irdm/slides/irdm-5-3.pdf

(Bruk på eget ansvar, jeg bare snublet over den randomly)

((Men jeg ville skummet den ifbm oblig C))

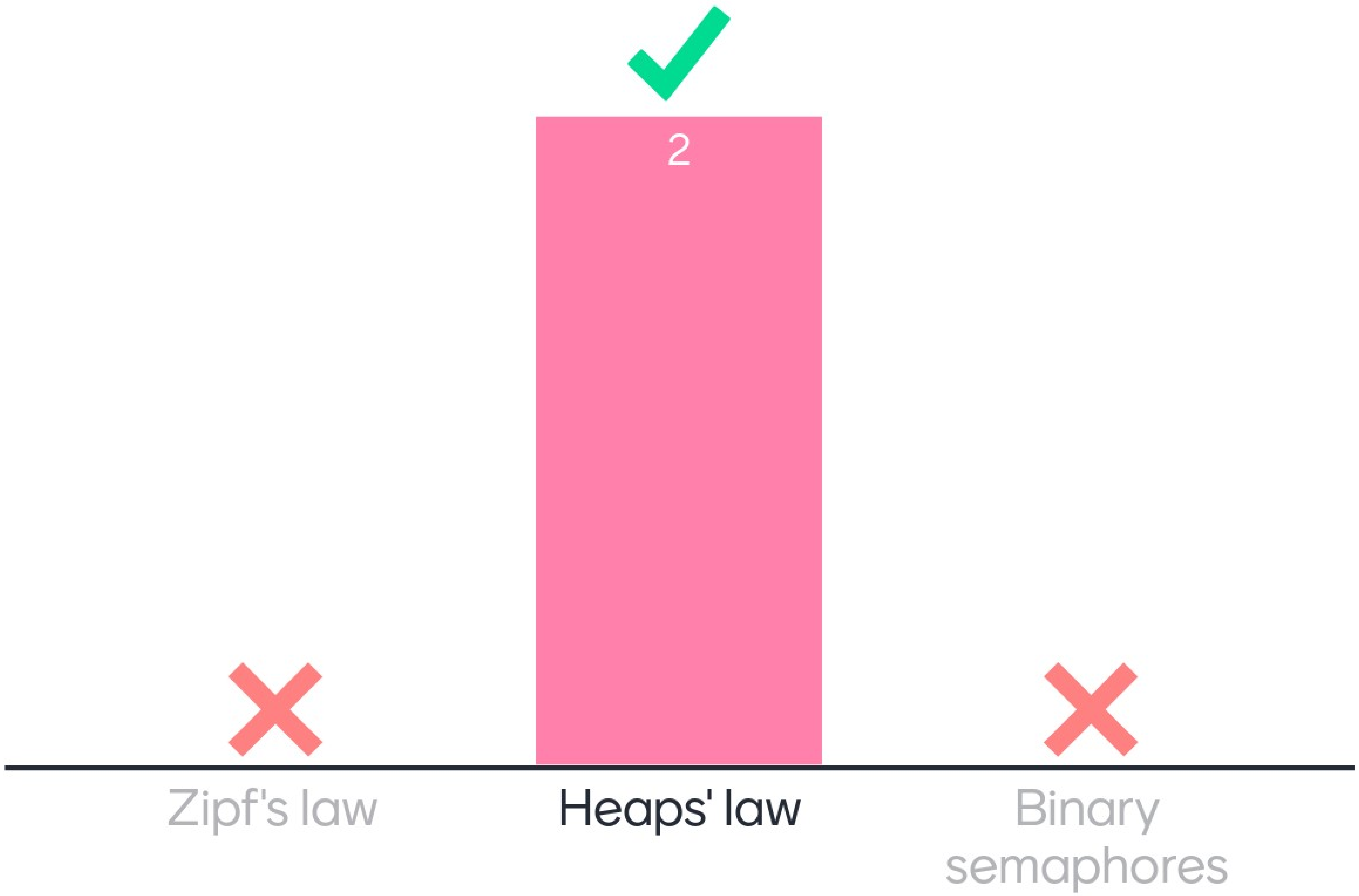
Menti-oppgaver!!!

Stjålet fra div eksamener

Burde være gjennomførbare

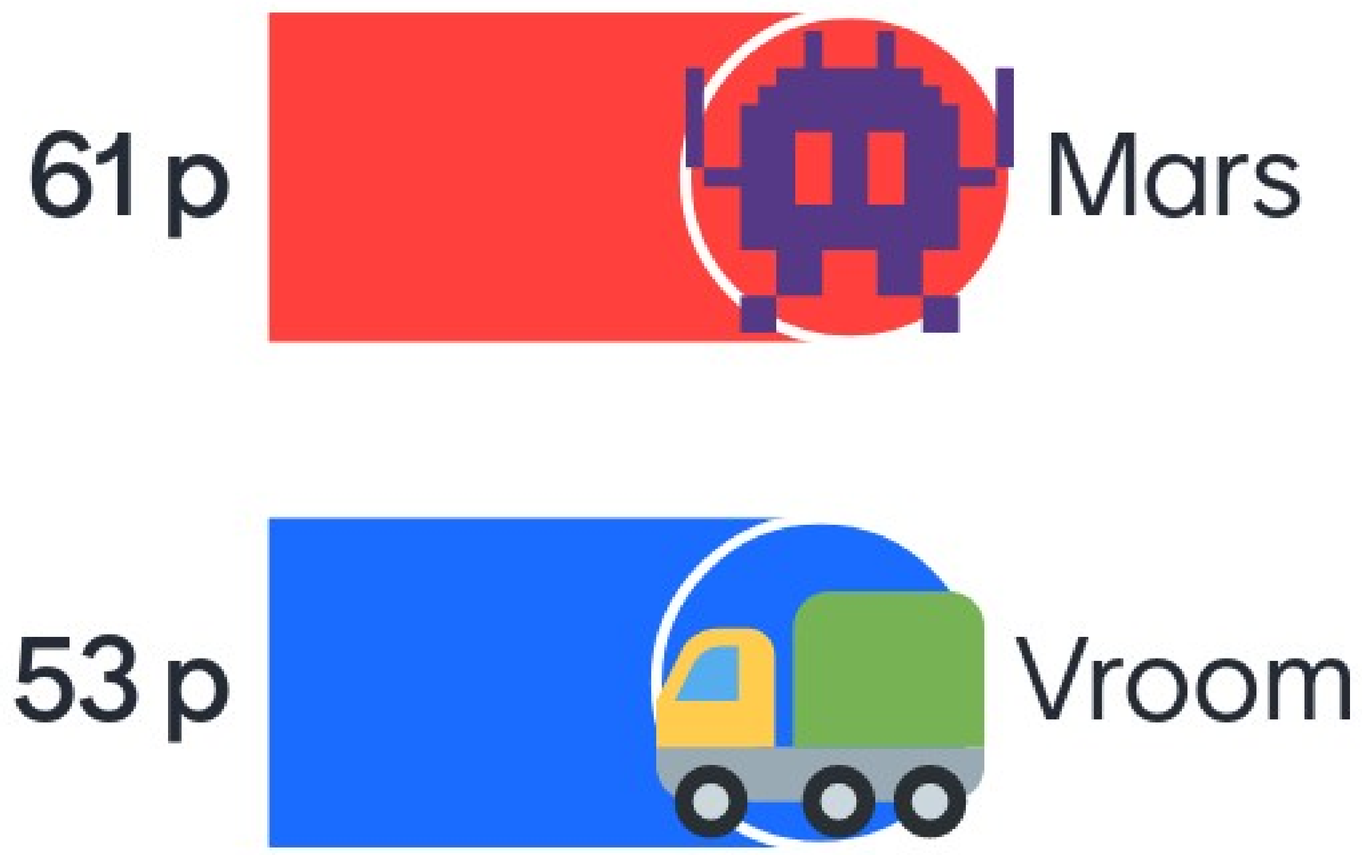
Hva kan man bruke for å finne ut av dette?

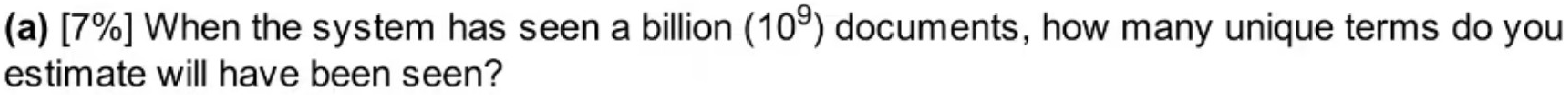
Mentimeter



2

Leaderboard

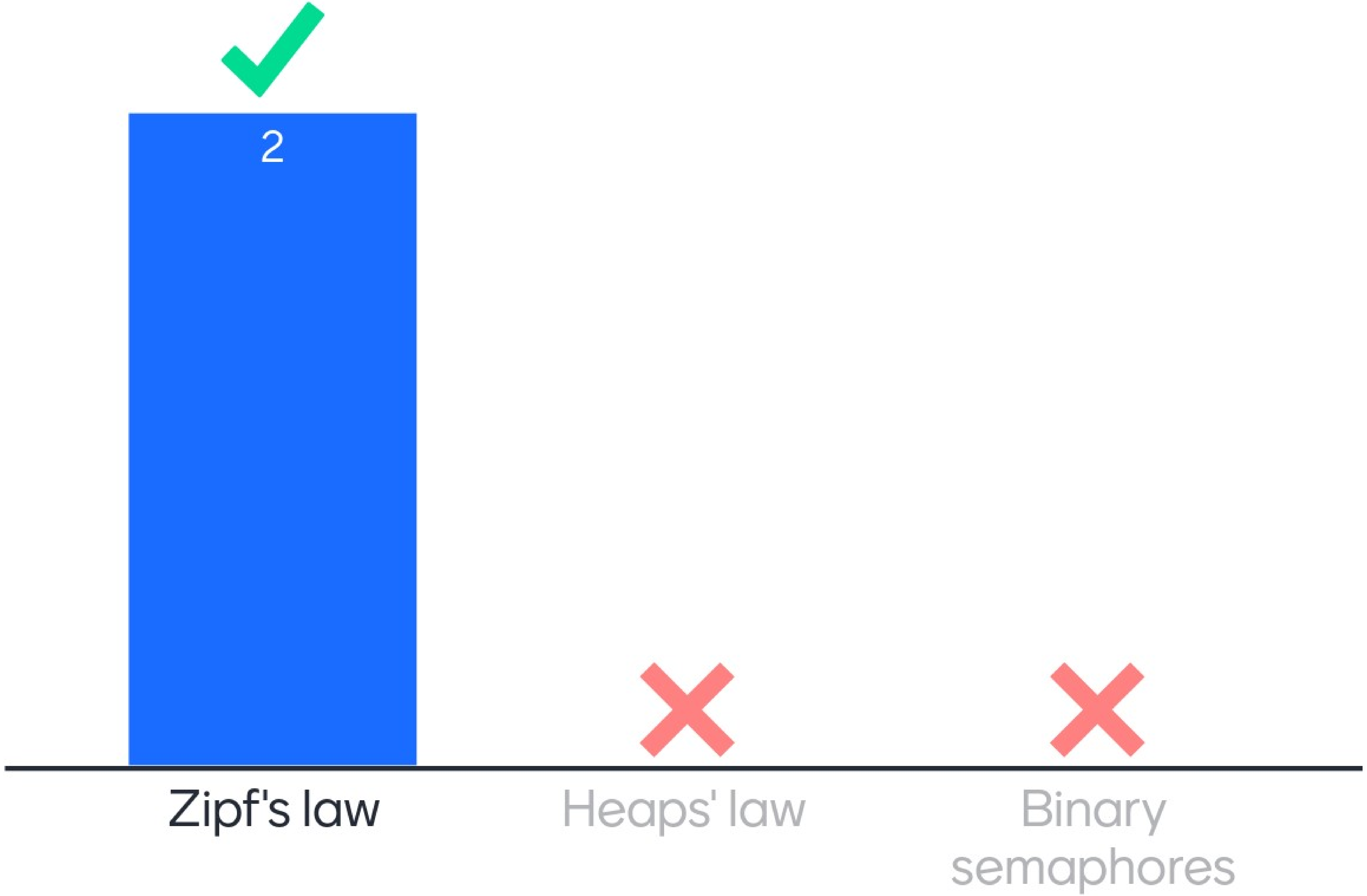




(a) [7%] When the system has seen a billion (10^9) documents, how many unique terms do you estimate will have been seen?

"Heaps' law (also called Herdan's law) is an empirical law which describes the number of distinct words in a document (or set of documents)"

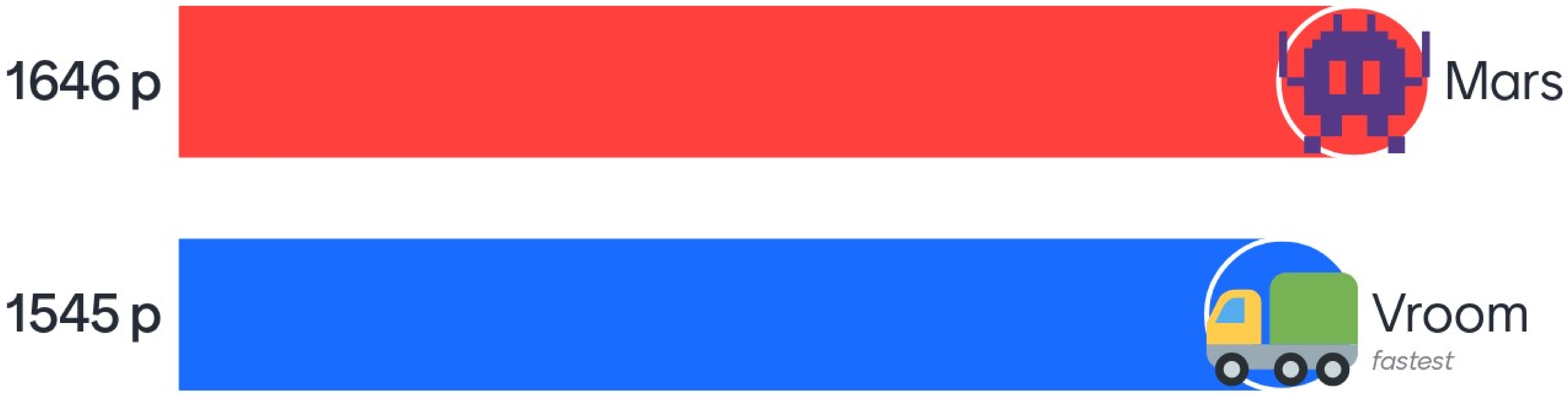
Hva kan man bruke for å finne ut av dette?



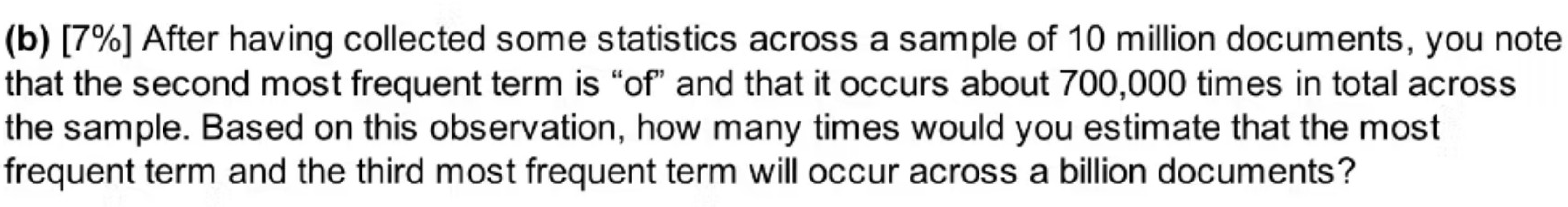
2

Mentimeter

Leaderboard



2



(b) [7%] After having collected some statistics across a sample of 10 million documents, you note that the second most frequent term is “of” and that it occurs about 700,000 times in total across the sample. Based on this observation, how many times would you estimate that the most frequent term and the third most frequent term will occur across a billion documents?

Zipf's law -> det mest frekvente ordet finnes 2x så ofte som det nest mest frekvente ordet, 3x så ofte som det tredje mest frekvente, osv. (NB: typo)

Noen tanker? Mange riktige svar

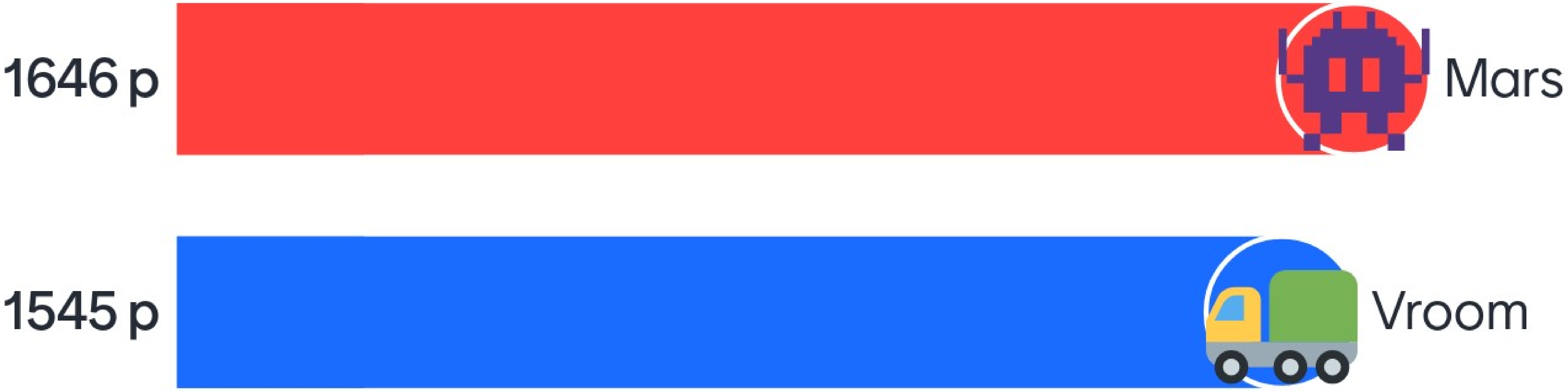
tar lang tid å decode, ikke lossless? ✕

The correct answer is: slow

1

Leaderboard

Mentimeter



1

Oblighjelp

Siste inspurt for oblig B - frist på fredag

Husk WCFD!!

(Husk også at 'spørre om hjelp' > plagiere)