Bloom filtr

- chceme udržovat $S \subseteq U$, ale v prostoru $o(|S|)$
- budeme opět hashovat, ale hash tabulka bude mít velikost $o(|S|)$
- a připouštíme false positive

  pro $x \in S$ $\Pr[$ alg vrátí „$x \in S$“ $] = 1$

  pro $x \notin S$ $\Pr[$ ——— $] > 0$

- máme m-bitové pole $A[0, \dots, m-1]$ zpočátku vynulované
- budeme používat k nezávislých hash. fcí $h_1, \dots, h_k$
  - dnes opět zcela náhodné ☺

Insert (x):

1. pro $i \leftarrow 1, \dots, k$
   2. $A[h_i(x)] \leftarrow 1$

Find (x)

1. return $A[h_1(x)] \wedge \dots \wedge A[h_k(x)]$

Zahashovali jsme n prvků (tedy $n = 191$).

$Pr[A[i] = 0]$ pro nějaké i?

- $\left(\left(1 - \frac{1}{m}\right)^{k}\right)^{n}$
  - $\left(1 - \frac{1}{m}\right)^{k}$: každá hash fce netrefí $A[i]$
  - $(\ldots)^{n}$: to nastane pro všech n prvků
  - zahashuj se kamkoliv jinam
- využijeme $m >> k$ a $\lim_{a \to \infty} \left(1 - \frac{1}{a}\right)^{a} = e$

$\rightarrow Pr[A[:] = 0] \approx e^{-\frac{kn}{m}}$

Pst false positive?

- pro prvek $x \notin Y$ musí platit, že $A[h_1(x)], \ldots, A[h_k(x)]$ musí všechny být 1
- pst, že nejsem 1 je z předchozího výpočtu $\approx e^{-\frac{kn}{m}}$
- $Pr[\text{false positive}] \approx \left(1 - e^{-\frac{kn}{m}}\right)^{k}$ ← pro $h_1, h_2, \ldots, h_k$
  - doplněk $A[h_i(x)] = 0$, tedy $A[h_i(x)] = 1$

Chceme mít pst. false positive pro pevné Y menší než ε pro $\varepsilon < 1$. Jak vybrat m a k?

- označ $p = e^{-\frac{kn}{m}}$
- $Pr[\text{false positive}] \leq (1-p)^{k}$
- označ $f = (1-p)^{k}$

- představme si, že m je taky první
- chceme minimalizovat fci f v prom. k
- ozn $g = \ln f = k \ln(1-p)$
- $\frac{dg}{dk} = \ln(1-p) + \frac{\ln}{m} \cdot \frac{e^{-\frac{k \ln}{m}}}{1 - e^{-\frac{k \ln}{m}}}$
- ověřte si, že glob. min je v bodě $k = \ln 2 \cdot \frac{n}{m}$
- např. pro $m = \frac{n}{2}$ chceme $k = 2 \ln 2 = 1.39 \leq 2$

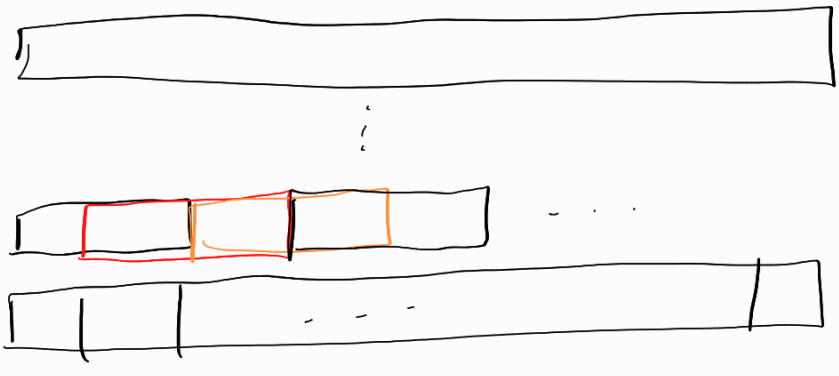
SUFFIX ARRAY

- pro řetězec S délky n máme pole A délky n
- $A[i]$ ... index i-tého nejmenšího suffixu S
- $L[i]$ ... délka nejdelšího společného prefixu $S[A[i]:]$ a $S[A[i+1]:]$

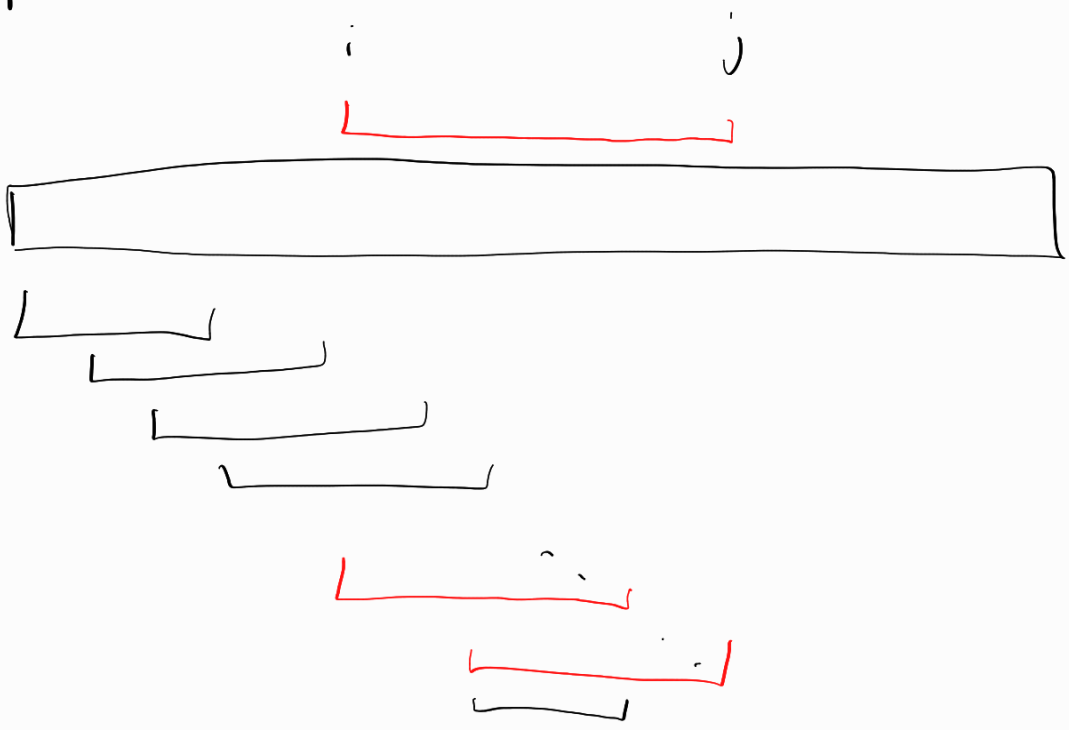
Jak spočítat nejdelší společný prefix $S[A[i]:]$ a $S[A[j]:]$, pokud máme pole L?

RMQ

- = range min query
- máme pole A délky n
- chceme DS pro rychlé dotazy „min z $A[i], A[i+1], \dots, A[j]$"



- $k = j - i + 1$
- spočítáme p t.ž. $2^p \leq k$ a $2^{p+1} \geq k$
- podíváme se na hladiny pro intervaly délky 2^p



- bereme min z červených intervalů

↪ 2 dotazy ... konst. složitost

- sice se nám překrývají kus, ale minimu to nevadí
- funguje pro max, gcd, ...
- obecně pro operace, kterým nevadí opakovaná aplikace na prvky → aby nevadil ten překryv

Jak za pomoci L hledat výskyty patternu P v čase $O(n + \lg m)$?

- opět binární půlení
- l ... zarážka vlevo

  r ... zarážka vpravo

  m ... střed
- ozn. $s(i) = T[SA[i]:]$
- představme si, že z rekurze známe $lcp(s(l), P)$ a $lcp(P, s(r))$
- pomocí sparse table v konst. čase spočítáme $lcp(s(l), s(m))$ a $lcp(s(m), s(r))$
- 3 případy, ozn. $k \leftarrow lcp(s(l), P)$

  i) $k < lcp(s(l), s(m))$

  . $s(l)$ a $s(m)$ mají delší spol. prefix než $s(m)$ a P

  $\rightarrow s(l)[k+1] = s(m)[k+1]$ a $s(l) \leq_{lex} P$

  . tím pádem víc než $s(P)[k+1] > s(m)[k+1]$

  . můžeme tedy nastavit $L \leftarrow m$

  ii) $k > lcp(s(l), s(m))$

  . tím pádem $P <_{lex} s(m)$

  . $r \leftarrow m$

ii.) $k = lcp(s(l), s(m))$

- pak podle prvního rozdílného znaku $P[k:]$ a $s(m)[k:]$ zjistíme, jestli $P \leq_{lex} s(m)$ nebo naopak

- jeden znak P nikdy nesrovnáváme s $s(m)$ dvakrát