

# Bulov model

Dragan Ivanović  
dragan.ivanovic@uns.ac.rs

Katedra za informatiku, Fakultet tehničkih nauka, Novi Sad

2015.

# Bulov model

- Zasnovan na teoriji skupova i Bulovoj algebri
- Posmatrani pojam se ili nalazi ili ne nalazi u dokumentu
- Nema rangiranja
- Nema parcijalnog poklapanja upita i dokumenta
- Konjukcija tri terma: jednako se posmatra i dokument koji nema ni jedan term i dokument koji ima dva terma

# Bulov model

- Tražene informacije se izražavaju upitom
- Upiti su logički izrazi, npr. Caesar AND Brutus
- Pretraživač vraća sve dokumente koji zadovoljavaju logički izraz

Da li Google koristi Bulov model?

# Bulovi upiti

- Bulov model može odgovoriti na svaki upit koji je Bulov izraz
  - Bulovi upiti su upiti koji koriste operatore AND, OR i NOT za kombinovanje termova u upitu.
  - posmatra svaki dokument kao **skup** termova.
  - nedvosmislen: dokument ili zadovoljava upit ili ne
- Osnovni mehanizam za pretraživanje preko 30 godina
- Korisnici (npr. advokati) i dalje vole da ga koriste
  - tačno se zna sta je rezultat
- Koristi se u mnogo različitih sistema (npr. email)

# Westlaw: primer Bulovog pretraživača

- Najveći komercijalni pretraživač pravnih materijala u smislu broja pretplatnika
- Preko pola miliona pretplatnika postavlja milione upita dnevno nad desetinama terabajta teksta
- Servis je pokrenut 1975.
- U 2005. Bulovi upiti (“Terms and Connectors”) su i dalje podrazumevani tip upita, i koristi ga veliki procenat korisnika
- ...iako rangirana pretraga postoji od 1992.

# Westlaw: primeri upita

*Zahtev:* Informacije o pravnoj teoriji o sprečavanju odavanja poslovnih tajni od strane zaposlenih koji su prethodno bili zaposleni u konkurentskoj firmi

*Upit:* "poslovna tajna" /s oda! /s spreč! /s zaposle!

*Zahtev:* Uslovi za osobe sa posebnim potrebama kod pristupa radnom mestu

*Upit:* posebn! /s potreb! /p pristup /s radnom mestu (zaposlen /4 mesto)

*Zahtev:* Slučajevi o odgovornosti domaćina prema pijanom gostu

*Upit:* domaćin! /p odgovor! /p pijan! /p gost!

# Westlaw: komentari

- $/4$  = u okviru četiri reči
- Razmak je disjunkcija, a ne konjunkcija! (konvencija u pre-Google eri)
- Dugački precizni upiti: operatori blizine, inkrementalni razvoj upita, različito od pretrage na webu
- Zašto profesionalni korisnici vole Bulov model: preciznost, kontrola, transparentnost
- Kada je Bulov model najbolji za pretragu? Zavisi od potreba korisnika, kolekcije dokumenata, veštine korisnika ...

# Bulov model

- Koje doktorske disertacije Univerziteta u Novom Sadu sadrže reči Lucene i pretrage, ali ne MULTIMEDIJALNIH?
- Možemo uraditi **grep** nad svim doktorskim disertacijama tražeći Lucene i pretrage, i onda odbaciti sve linije koje sadrže multimedijalnih.
- Zašto ovo nije dobro rešenje?



# Bulov model

- Koje doktorske disertacije Univerziteta u Novom Sadu sadrže reči Lucene i pretrage, ali ne MULTIMEDIJALNIH?
- Možemo uraditi **grep** nad svim doktorskim disertacijama tražeći Lucene i pretrage, i onda odbaciti sve linije koje sadrže multimedijalnih.
- Zašto ovo nije dobro rešenje?
  - sporo (za velike kolekcije)
  - „NOT multimedijalnih“ nije trivijalno
  - druge operacije (npr. nađi reč Lucene **blizu** reči pretrage) nisu izvodljive
  - rangiranje pogodaka ne postoji

# Matrica incidencije term/dokument

	Ivanović	Milosavljević	Gostojić	Zarić	...
	D.	B.	S.	M.	...
digitalan	1	1	1	1	
lucene	1	1	1	1	
dokument	1	1	1	1	
obrazovanje	0	0	1	0	
pretraga	1	1	1	1	
multimedijalan	0	1	1	1	
evaluacija	0	0	0	0	

...

Vrednost je 1 ako se term pojavljuje u dokumentu. Primer: obrazovanje se javlja u disertaciji *Gostojić S.* Vrednost je 0 ako se term ne pojavljuje u dokumentu. Primer: obrazovanje se ne javlja u *Milosavljević B.*

# Matrica incidencije term/dokument

	Ivanović	Milosavljević	Gostojić	Zarić	...
	D.	B.	S.	M.	...
digitalan	1	1	1	1	
lucene	1	1	1	1	
dokument	1	1	1	1	
obrazovanje	0	0	1	0	
pretraga	1	1	1	1	
multimedijalan	0	1	1	1	
evaluacija	0	0	0	0	

...

Vrednost je 1 ako se term pojavljuje u dokumentu. Primer: obrazovanje se javlja u disertaciji *Gostojić S.* Vrednost je 0 ako se term ne pojavljuje u dokumentu. Primer: obrazovanje se ne javlja u *Milosavljević B.*

# Matrica incidencije term/dokument

	Ivanović	Milosavljević	Gostojić	Zarić	...
	D.	B.	S.	M.	...
digitalan	1	1	1	1	
lucene	1	1	1	1	
dokument	1	1	1	1	
obrazovanje	0	0	1	0	
pretraga	1	1	1	1	
multimedijalan	0	1	1	1	
evaluacija	0	0	0	0	

...

Vrednost je 1 ako se term pojavljuje u dokumentu. Primer: obrazovanje se javlja u disertaciji *Gostojić S.* Vrednost je 0 ako se term ne pojavljuje u dokumentu. Primer: obrazovanje se ne javlja u *Milosavljević B.*

# Vektor incidencije

- Postoji vektor sa elementima 0/1 za svaki term
- Za upit Lucene AND pretrage AND NOT multimedijalnih:
  - pretprocesiranje upita da bi se tokeni iz upita prebacili u termove: lucene AND pretraga AND NOT multimedijalan,
  - uzimanje vektora incidencije za lucene (*1111*), pretraga (*1111*) i multimedijalan (*0111*),
  - izračunavanje komplementa za vektor incidencije za multimedijalan (*1000*),
  - izračunavanje logičkog I (AND) po bitovima (eng. BITWISE AND) za tri vektora *1111* AND *1111* AND *1000* = *1000*

# Odgovor na upit

*Ivanović D., Informacioni sistem naučno-istraživačke delatnosti*  
...IndexSearcher klasa pripada biblioteci Apache **Lucene**. Ova klasa je namenjena **pretraži** dokumenata koji su prethodno indeksirani upotrebom biblioteke Apache **Lucene**...

# Veće kolekcije

- Ako kolekcija ima  $N = 10^6$  dokumenata, svaki sa 1000 tokena

# Veće kolekcije

- Ako kolekcija ima  $N = 10^6$  dokumenata, svaki sa 1000 tokena
- U proseku 6 bajtova po tokenu, uključujući razmake i interpunkciju  $\Rightarrow$  veličina kolekcije je oko 6 GB



# Veće kolekcije

- Ako kolekcija ima  $N = 10^6$  dokumenata, svaki sa 1000 tokena
- U proseku 6 bajtova po tokenu, uključujući razmake i interpunkciju  $\Rightarrow$  veličina kolekcije je oko 6 GB
- Pretpostavimo da ima  $M = 500,000$  različitih termova u kolekciji

# Veće kolekcije

- Ako kolekcija ima  $N = 10^6$  dokumenata, svaki sa 1000 tokena
- U proseku 6 bajtova po tokenu, uključujući razmake i interpunkciju  $\Rightarrow$  veličina kolekcije je oko 6 GB
- Pretpostavimo da ima  $M = 500,000$  različitih termova u kolekciji
- (Pravimo razliku između terma i tokena)

# Ogromna matrica incidencije

- $M = 500,000 \times 10^6 =$  pola biliona nula i jedinica
- A matrica ne sadrži više od milion jedinica
  - matrica je vrlo retka
- Ima li bolje reprezentacije?
  - pamtimo samo jedinice

# Invertovani indeks

Za svaki term  $t$ , čuvamo listu dokumenata koji sadrže  $t$ .

lucene	→	2	31	54	101
--------	---	---	----	----	-----

pretraga	→	1	2	4	5	6	16	57	132	...
----------	---	---	---	---	---	---	----	----	-----	-----

multimedijalan	→	1	2	4	11	31	45	173	174
----------------	---	---	---	---	----	----	----	-----	-----

⋮

rečnik

pojave

# Konstrukcija invertovanog indeksa

- 1 Prikupljanje dokumenata koje treba indeksirati:

Lucene biblioteka je otvorenog koda.

Koristi se za pretragu tekstualnih sadržaja. . . . ,

- 2 Tokenizacija teksta - pretvaranje svakog dokumenta u listu tokena:

Lucene biblioteka je otvorenog . . . ,

- 3 Pretprocesiranje teksta - formiranje liste normalizovanih tokena, tj. termova koji će biti u rečniku:

lucene biblioteka biti otvoren . . . ,

- 4 Indeksiranje dokumenata - formiranje invertovanog indeksa koji ima rečnik i pojave.

# Tokenizacija i pretprocesiranje

**Dokument 1.** Apache Lucene je javno dostupna biblioteka pisana u Javi namenjena pretraživanju teksta

**Dokument 2.** Program-ska biblioteka Apache Lucene omogućava full-text pretraživanje sa rangiranjem rezultata



**Dokument 1.** apache lucene biti javno dostupan biblioteka pisan u java namenjen pretrazivanje tekst

**Dokument 2.** program-ski biblioteka apache lucene omoguciti full text pretrazivanje sa rangiranje rezultat

# Izračunavanje pojava

**Dokument 1.** apache lucene biti javan dostupan biblioteka pisan u java namenjen pretrazivanje tekst

**Dokument 2.** programski biblioteka apache lucene omoguciti full text pretrazivanje sa rangiranje rezultat



term	docID
apache	1
lucene	1
biti	1
javan	1
dostupan	1
biblioteka	1
pisan	1
u	1
java	1
namenjen	1
pretrazivanje	1
tekst	1
programski	2
biblioteka	2
apache	2
lucene	2
omoguciti	2
full	2
text	2
pretrazivanje	2
sa	2
rangiranje	2
rezultat	2

# Sortiranje pojava

term	docID		term	docID
apache	1		apache	1
lucene	1		apache	2
biti	1		biblioteka	1
javan	1		biblioteka	2
dostupan	1		biti	1
biblioteka	1		dostupan	1
pisan	1		full	2
u	1		java	1
java	1		javan	1
namenjen	1		lucene	1
pretrazivanje	1	⇒	lucene	2
tekst	1		namenjen	1
programski	2		omoguciti	2
biblioteka	2		pisan	1
apache	2		pretrazivanje	1
lucene	2		pretrazivanje	2
omoguciti	2		programski	2
full	2		rangiranje	2
text	2		rezultat	2
pretrazivanje	2		sa	2
sa	2		tekst	1
rangiranje	2		text	2
rezultat	2		u	1



# Kreiranje liste pojava, određivanje frekvencije pojavljivanja

term	docID	term	frekv.	→	liste pojava
apache	1	apache	2	→	1 → 2
apache	2	biblioteka	2	→	1 → 2
biblioteka	1	biti	1	→	1
biblioteka	2	dostupan	1	→	1
biti	1	full	1	→	2
dostupan	1	java	1	→	1
full	2	javan	1	→	1
java	1	lucene	2	→	1 → 2
javan	1	namenjen	1	→	1
lucene	1	omoguciti	1	→	2
lucene	2	pisan	1	→	1
namenjen	1	pretrazivanje	2	→	1 → 2
omoguciti	2	programski	1	→	2
pisan	1	rangiranje	1	→	2
pretrazivanje	1	rezultat	1	→	2
pretrazivanje	2	sa	1	→	2
programski	2	tekst	1	→	1
rangiranje	2	text	1	→	2
rezultat	2	u	1	→	1
sa	2				
tekst	1				
text	2				
u	1				



rečnik

pojave

# Implementacija invertovanog indeksa - pitanja/dileme

- Konstrukcija: kako kreirati indeks za ogromne kolekcije?
- Koliko memorije nam treba za rečnik i indeks?
- Kompresija indeksa: kako efikasno skladištiti i obrađivati indeks za velike kolekcije?
- Rangiranje rezultata: kako izgleda invertovani indeks kada nam treba „najbolji“ odgovor?

# Jednostavan konjunktivni upit (dva terma)

- Razmotrimo jednostavan konjunktivni upit dva tokena Lucene AND multimedijalnih
- Algoritam za pronalaženje svih relevantnih dokumenata za ovaj upit pomoću invertovanog indeksa kreiranog na prethodno opisani način je sledeći:
  - 1 pretprocesiranje upita nakon čega se dobija konjuktivni upit dva terma (ne dva tokena, nego dva terma): lucene AND multimedijalan,
  - 2 pronalaženje terma lucene u rečniku termova,
  - 3 učitavanje liste pojava ovog terma iz fajla sa pojavama,
  - 4 pronalaženje terma multimedijalan u rečniku termova,
  - 5 učitavanje liste pojava ovog terma iz fajla sa pojavama,
  - 6 izračunavanje **preseka** ove dve liste pojava,
  - 7 vraćanje rezultata korisniku - vraćanje dokumenata koji se nalazu u prethodno izračunatom preseku.

## Izračunavanje preseka dve liste pojava: algoritam

```
Intersect( $p_1, p_2$ )  
1   $answer \leftarrow \langle \rangle$   
2  while  $p_1 \neq \text{nil}$  and  $p_2 \neq \text{nil}$   
3  do if  $\text{docID}(p_1) = \text{docID}(p_2)$   
4      then  $\text{Add}(answer, \text{docID}(p_1))$   
5           $p_1 \leftarrow \text{next}(p_1)$   
6           $p_2 \leftarrow \text{next}(p_2)$   
7      else if  $\text{docID}(p_1) < \text{docID}(p_2)$   
8          then  $p_1 \leftarrow \text{next}(p_1)$   
9          else  $p_2 \leftarrow \text{next}(p_2)$   
10 return  $answer$ 
```

## Izračunavanje preseka dve liste pojava

lucene  $\longrightarrow$   $\boxed{2} \rightarrow \boxed{31} \rightarrow \boxed{54} \rightarrow \boxed{101}$

multimedijalan  $\longrightarrow$   $\boxed{1} \rightarrow \boxed{2} \rightarrow \boxed{4} \rightarrow \boxed{11} \rightarrow \boxed{31} \rightarrow \boxed{45} \rightarrow \boxed{173} \rightarrow \boxed{174}$

**presek**  $\implies$

## Izračunavanje preseka dve liste pojava

lucene  $\longrightarrow$  2  $\rightarrow$  31  $\rightarrow$  54  $\rightarrow$  101

multimedijalan  $\longrightarrow$  1  $\rightarrow$  2  $\rightarrow$  4  $\rightarrow$  11  $\rightarrow$  31  $\rightarrow$  45  $\rightarrow$  173  $\rightarrow$  174

**presek**  $\implies$

## Izračunavanje preseka dve liste pojava

lucene  $\longrightarrow$  2  $\rightarrow$  31  $\rightarrow$  54  $\rightarrow$  101

multimedijalan  $\longrightarrow$  1  $\rightarrow$  2  $\rightarrow$  4  $\rightarrow$  11  $\rightarrow$  31  $\rightarrow$  45  $\rightarrow$  173  $\rightarrow$  174

**presek**  $\implies$

## Izračunavanje preseka dve liste pojava

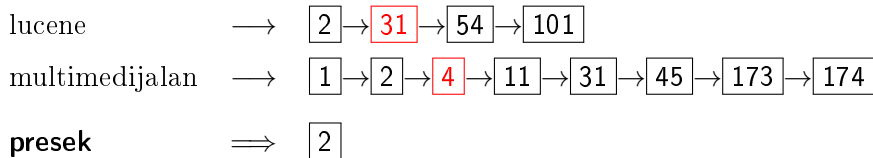
lucene  $\longrightarrow$  2  $\rightarrow$  31  $\rightarrow$  54  $\rightarrow$  101

multimedijalan  $\longrightarrow$  1  $\rightarrow$  2  $\rightarrow$  4  $\rightarrow$  11  $\rightarrow$  31  $\rightarrow$  45  $\rightarrow$  173  $\rightarrow$  174

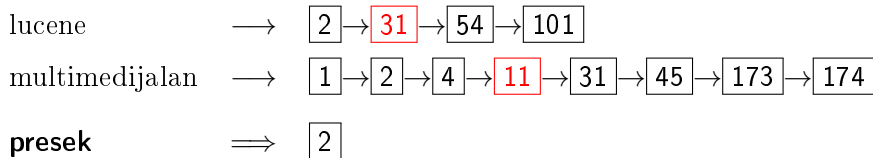
**presek**  $\implies$  2



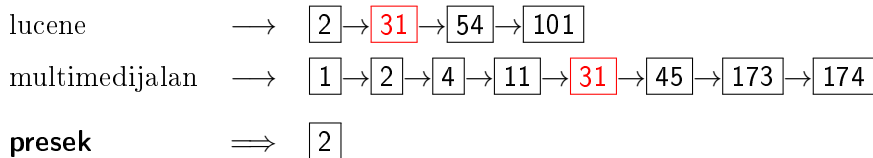
## Izračunavanje preseka dve liste pojava



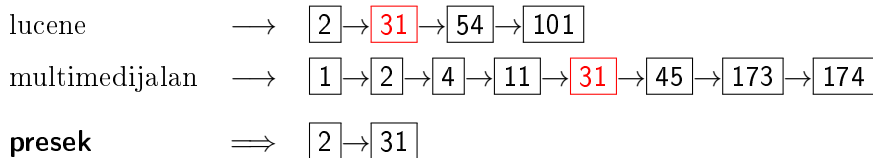
## Izračunavanje preseka dve liste pojava



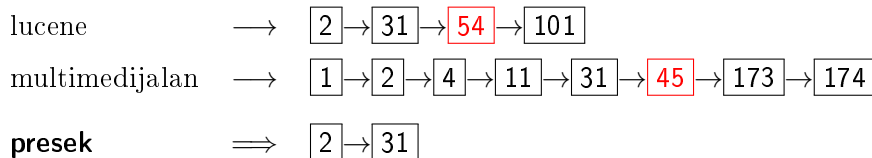
## Izračunavanje preseka dve liste pojava



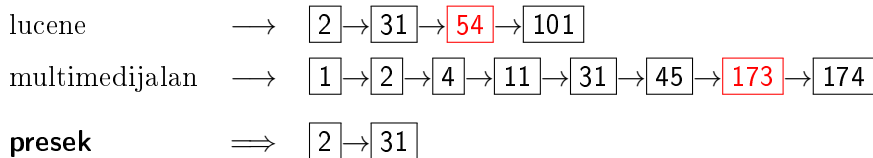
## Izračunavanje preseka dve liste pojava



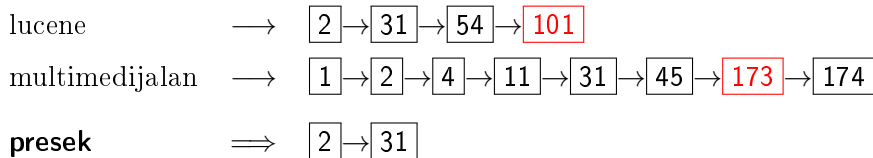
# Izračunavanje preseka dve liste pojava



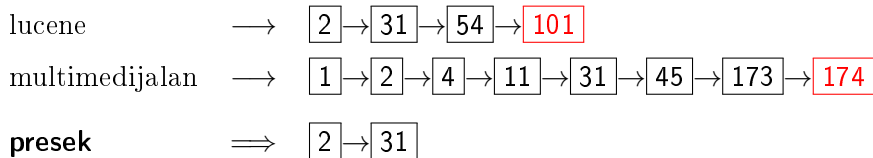
## Izračunavanje preseka dve liste pojava



## Izračunavanje preseka dve liste pojava



## Izračunavanje preseka dve liste pojava





# Izračunavanje preseka dve liste pojava

lucene  $\longrightarrow$   $\boxed{2} \rightarrow \boxed{31} \rightarrow \boxed{54} \rightarrow \boxed{101}$

multimedijalan  $\longrightarrow$   $\boxed{1} \rightarrow \boxed{2} \rightarrow \boxed{4} \rightarrow \boxed{11} \rightarrow \boxed{31} \rightarrow \boxed{45} \rightarrow \boxed{173} \rightarrow \boxed{174}$

**presek**  $\implies$   $\boxed{2} \rightarrow \boxed{31}$

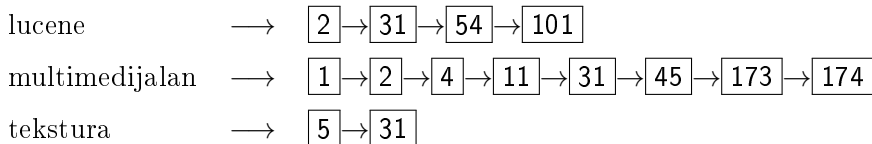
- linearna zavisnost od dužine liste
- radi samo ako su liste sortirane

# Optimizacija upita

- Koja je minimalna složenost za implementaciju upita?
- Posmatramo upit koji je konjunkcija (AND) od  $n$  termova,  $n > 2$
- Za svaki term, pribavi njegovu listu pojava, onda izračunaj AND za sve njih

# Optimizacija upita

- Primer upita: lucene AND multimedijalan AND tekstura



# Optimizacija upita

- Primer upita: lucene AND multimedijalan AND tekstura
- jednostavna i efikasna optimizacija: obradi u rastućem redosledu frekvencije
- Počni od najkraće liste pojava
- U ovom primeru: prvo tekstura, potom lucene, potom multimedijalan

lucene → 

2
---

 → 

31
----

 → 

54
----

 → 

101
-----

multimedijalan → 

1
---

 → 

2
---

 → 

4
---

 → 

11
----

 → 

31
----

 → 

45
----

 → 

173
-----

 → 

174
-----

tekstura → 

5
---

 → 

31
----

# Optimizovani algoritam za konjunktivne upite

Intersect( $\langle t_1, \dots, t_n \rangle$ )

```
1  terms  $\leftarrow$  SortByIncreasingFrequency( $\langle t_1, \dots, t_n \rangle$ )
2  result  $\leftarrow$  postings(first(terms))
3  terms  $\leftarrow$  rest(terms)
4  while terms  $\neq$  nil and result  $\neq$  nil
5  do result  $\leftarrow$  Intersect(result, postings(first(terms)))
6     terms  $\leftarrow$  rest(terms)
7  return result
```

# Opštija optimizacija

- Primer upita:  
(lucene OR sphinx) AND (multimedijalan OR slika) AND  
(tekstura OR šablon)
- Pribavi frekvencije za sve termine
- Proceni veličinu svakog OR-a kao zbir frekvencija operanada  
(konzervativni pristup)
- Obradi u rastućem redosledu veličine OR-ova

# Osnovni algoritam za izračunavanje preseka

lucene  $\longrightarrow$   $\boxed{2} \rightarrow \boxed{31} \rightarrow \boxed{54} \rightarrow \boxed{101}$

multimedijalan  $\longrightarrow$   $\boxed{1} \rightarrow \boxed{2} \rightarrow \boxed{4} \rightarrow \boxed{11} \rightarrow \boxed{31} \rightarrow \boxed{45} \rightarrow \boxed{173} \rightarrow \boxed{174}$

**presek**  $\implies$   $\boxed{2} \rightarrow \boxed{31}$

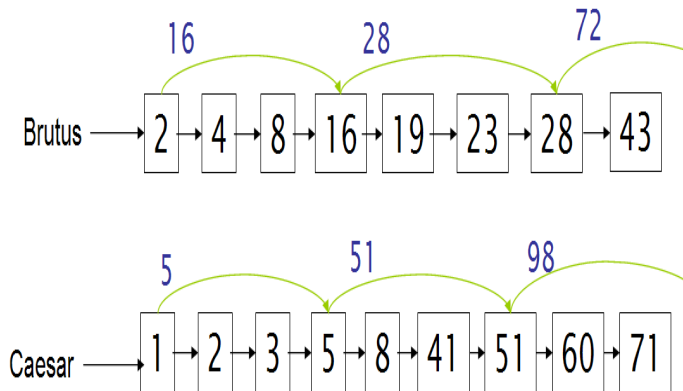
- Linearno zavisi od dužine lista
- Može li bolje?

# Pointeri za preskakanje (skip pointers)

- Skip pointeri omogućavaju **preskakanje** pojava koje svakako neće biti u rezultatu
- Izračunavanje preseka može da se ubrza na taj način
- Neke liste pojava sadrže više miliona elemenata – brzina može biti problem ako je algoritam linearan
- Gde da stavimo skip pointere?
- Tako da ne pokvarimo rezultat?



# Skip liste



## Izračunavanje preseka sa skip pointerima

IntersectWithSkips( $p_1, p_2$ )

```

1  answer  $\leftarrow \langle \rangle$ 
2  while  $p_1 \neq \text{nil}$  and  $p_2 \neq \text{nil}$ 
3  do if  $\text{docID}(p_1) = \text{docID}(p_2)$ 
4      then Add(answer,  $\text{docID}(p_1)$ )
5           $p_1 \leftarrow \text{next}(p_1)$ 
6           $p_2 \leftarrow \text{next}(p_2)$ 
7  else if  $\text{docID}(p_1) < \text{docID}(p_2)$ 
8      then if  $\text{hasSkip}(p_1)$  and  $(\text{docID}(\text{skip}(p_1)) \leq \text{docID}(p_2))$ 
9          then while  $\text{hasSkip}(p_1)$  and  $(\text{docID}(\text{skip}(p_1)) \leq \text{docID}(p_2))$ 
10             do  $p_1 \leftarrow \text{skip}(p_1)$ 
11             else  $p_1 \leftarrow \text{next}(p_1)$ 
12  else if  $\text{hasSkip}(p_2)$  and  $(\text{docID}(\text{skip}(p_2)) \leq \text{docID}(p_1))$ 
13      then while  $\text{hasSkip}(p_2)$  and  $(\text{docID}(\text{skip}(p_2)) \leq \text{docID}(p_1))$ 
14          do  $p_2 \leftarrow \text{skip}(p_2)$ 
15          else  $p_2 \leftarrow \text{next}(p_2)$ 
16  return answer

```

# Gde staviti skokove?

- Kompromis: Broj preskočenih elemenata / učestalost skakanja
- Više skokova: svaki skip pointer preskače malo elemenata, ali ga možemo češće koristiti
- Manje skokova: svaki skip pointer preskače puno elemenata, ali ga možemo retko koristiti

# Gde staviti skokove?

- Jednostavna heuristika: za liste pojava dužine  $P$ , napraviti  $\sqrt{P}$  skip pointera jednake dužine
- Ovo ignoriše distribuciju termova (podrazumeva uniformnu raspodelu)
- Jednostavno ako je indeks sporo promenljiv
- Koliko su skip pointeri korisni?
- Bili su jako korisni
- Sa današnjim procesorima nisu više presudni

# Upiti-fraze

- Treba da odgovorimo na upit "sive pantalone" – kao fraza (niz reči)
- Dakle, *Voli da nosi sive košulje i plave pantalone* nije pogodak
- Koncept fraze korisnici su brzo usvojili
- Oko 10% upita na webu su upiti-fraze
- Posledice za invertovani indeks: nije više dovoljno čuvati samo docID

# Dvorečni indeksi

- Indeksiraj svaki susedni par reči u tekstu kao frazu
- Na primer, *sive markirane pantalone* će dati dva para reči: “*sive markirane*” i “*markirane pantalone*”
- Svaki od parova se tretira kao term u rečniku
- Lako je odgovoriti na dvorečne upite

## Duži upiti-fraze

- Dugačka fraza kao "sive markirane pantalone" može da se prikaže kao  
"sive markirane" AND "markirane pantalone"
- Moramo da uradimo filtriranje pogodaka da izdvojimo samo one dokumente koji stvarno sadrže celu frazu: **Voli da nosi sive markirane košulje i plave markirane pantalone**

# Proširene dvoreči

- Parsiraj dokument i označi vrstu reči
- Vrste reči neka budu imenice (N) predlozi/prilozi (X)
- Tretiraj svaki niz reči oblika  $N \times N$  kao *proširenu dvoreč*
- Primeri:
 

Univerzitet	u	Beogradu
N	X	N
protokol	za	udaljenu pretragu
N	X	X N
- Uključi proširene dvoreči u rečnik



# Problemi sa dvorečnim indeksima

- False positive: pronađeni dokument koji zapravo ne sadrži celu dugačku frazu
- Eksplozija indeksa zbog velikog broja termova u rečniku

# Pozicioni indeksi

- Pozicioni indeksi su dobra zamena za dvorečne indekse
- Liste pojava u **nepozicionom** indeksu: svaka pojava je docID
- Liste pojava u **pozicionom** indeksu: svaka pojava je docID i **lista pozicija**
- Primer:  $to_1$   $be_2$   $or_3$   $not_4$   $to_5$   $be_6$

to, 993427:

$\langle 1, 6: \langle 7, 18, 33, 72, 86, 231 \rangle;$   
 $2, 5: \langle 1, 17, 74, 222, 255 \rangle;$   
 $4, 5: \langle 8, 16, 190, 429, 433 \rangle;$   
 $5, 2: \langle 363, 367 \rangle;$   
 $7, 3: \langle 13, 23, 191 \rangle; \dots \rangle$

be, 178239:

$\langle 1, 2: \langle 17, 25 \rangle;$   
 $4, 5: \langle 17, 191, 291, 430, 434 \rangle;$   
 $5, 3: \langle 14, 19, 101 \rangle; \dots \rangle$

# Pozicioni indeksi

- Pozicioni indeksi su dobra zamena za dvorečne indekse
- Liste pojava u **nepozicionom** indeksu: svaka pojava je docID
- Liste pojava u **pozicionom** indeksu: svaka pojava je docID i **lista pozicija**
- Primer:  $to_1$   $be_2$   $or_3$   $not_4$   $to_5$   $be_6$

to, 993427:

$\langle$  **1**, 6:  $\langle$  **7**, 18, 33, 72, 86, 231 $\rangle$ ;  
 2, 5:  $\langle$  1, 17, 74, 222, 255 $\rangle$ ;  
 4, 5:  $\langle$  8, 16, 190, 429, 433 $\rangle$ ;  
 5, 2:  $\langle$  363, 367 $\rangle$ ;  
 7, 3:  $\langle$  13, 23, 191 $\rangle$ ; ...  $\rangle$

be, 178239:

$\langle$  **1**, 2:  $\langle$  17, 25 $\rangle$ ;  
 4, 5:  $\langle$  17, 191, 291, 430, 434 $\rangle$ ;  
 5, 3:  $\langle$  14, 19, 101 $\rangle$ ; ...  $\rangle$

# Pozicioni indeksi

- Pozicioni indeksi su dobra zamena za dvorečne indekse
- Liste pojava u **nepozicionom** indeksu: svaka pojava je docID
- Liste pojava u **pozicionom** indeksu: svaka pojava je docID i **lista pozicija**
- Primer:  $to_1$   $be_2$   $or_3$   $not_4$   $to_5$   $be_6$

to, 993427:

$\langle$  **1**, 6:  $\langle$  **7**, 18, 33, 72, 86, 231 $\rangle$ ;  
 2, 5:  $\langle$  1, 17, 74, 222, 255 $\rangle$ ;  
 4, 5:  $\langle$  8, 16, 190, 429, 433 $\rangle$ ;  
 5, 2:  $\langle$  363, 367 $\rangle$ ;  
 7, 3:  $\langle$  13, 23, 191 $\rangle$ ; ...  $\rangle$

be, 178239:

$\langle$  **1**, 2:  $\langle$  **17**, 25 $\rangle$ ;  
 4, 5:  $\langle$  17, 191, 291, 430, 434 $\rangle$ ;  
 5, 3:  $\langle$  14, 19, 101 $\rangle$ ; ...  $\rangle$

# Pozicioni indeksi

- Pozicioni indeksi su dobra zamena za dvorečne indekse
- Liste pojava u **nepozicionom** indeksu: svaka pojava je docID
- Liste pojava u **pozicionom** indeksu: svaka pojava je docID i **lista pozicija**
- Primer:  $to_1$   $be_2$   $or_3$   $not_4$   $to_5$   $be_6$

to, 993427:

$\langle$  **1**, 6:  $\langle$ 7, **18**, 33, 72, 86, 231 $\rangle$ ;  
 2, 5:  $\langle$ 1, 17, 74, 222, 255 $\rangle$ ;  
 4, 5:  $\langle$ 8, 16, 190, 429, 433 $\rangle$ ;  
 5, 2:  $\langle$ 363, 367 $\rangle$ ;  
 7, 3:  $\langle$ 13, 23, 191 $\rangle$ ; ...  $\rangle$

be, 178239:

$\langle$  **1**, 2:  $\langle$ 17, 25 $\rangle$ ;  
 4, 5:  $\langle$ 17, 191, 291, 430, 434 $\rangle$ ;  
 5, 3:  $\langle$ 14, 19, 101 $\rangle$ ; ...  $\rangle$

# Pozicioni indeksi

- Pozicioni indeksi su dobra zamena za dvorečne indekse
- Liste pojava u **nepozicionom** indeksu: svaka pojava je docID
- Liste pojava u **pozicionom** indeksu: svaka pojava je docID i **lista pozicija**
- Primer:  $to_1$   $be_2$   $or_3$   $not_4$   $to_5$   $be_6$

to, 993427:

$\langle 1, 6: \langle 7, 18, 33, 72, 86, 231 \rangle;$   
 $2, 5: \langle 1, 17, 74, 222, 255 \rangle;$   
 $4, 5: \langle 8, 16, 190, 429, 433 \rangle;$   
 $5, 2: \langle 363, 367 \rangle;$   
 $7, 3: \langle 13, 23, 191 \rangle; \dots \rangle$

be, 178239:

$\langle 1, 2: \langle 17, 25 \rangle;$   
 $4, 5: \langle 17, 191, 291, 430, 434 \rangle;$   
 $5, 3: \langle 14, 19, 101 \rangle; \dots \rangle$

# Pozicioni indeksi

- Pozicioni indeksi su dobra zamena za dvorečne indekse
- Liste pojava u **nepozicionom** indeksu: svaka pojava je docID
- Liste pojava u **pozicionom** indeksu: svaka pojava je docID i **lista pozicija**
- Primer:  $to_1$   $be_2$   $or_3$   $not_4$   $to_5$   $be_6$

to, 993427:

$\langle$  **1**, 6:  $\langle$ 7, 18, **33**, **72**, **86**, **231** $\rangle$ ;  
 2, 5:  $\langle$ 1, 17, 74, 222, 255 $\rangle$ ;  
 4, 5:  $\langle$ 8, 16, 190, 429, 433 $\rangle$ ;  
 5, 2:  $\langle$ 363, 367 $\rangle$ ;  
 7, 3:  $\langle$ 13, 23, 191 $\rangle$ ; ... $\rangle$

be, 178239:

$\langle$  **1**, 2:  $\langle$ 17, 25 $\rangle$ ;  
 4, 5:  $\langle$ 17, 191, 291, 430, 434 $\rangle$ ;  
 5, 3:  $\langle$ 14, 19, 101 $\rangle$ ; ... $\rangle$

# Pozicioni indeksi

- Pozicioni indeksi su dobra zamena za dvorečne indekse
- Liste pojava u **nepozicionom** indeksu: svaka pojava je docID
- Liste pojava u **pozicionom** indeksu: svaka pojava je docID i **lista pozicija**
- Primer:  $to_1$   $be_2$   $or_3$   $not_4$   $to_5$   $be_6$

to, 993427:

$\langle 1, 6: \langle 7, 18, 33, 72, 86, 231 \rangle;$   
 $2, 5: \langle 1, 17, 74, 222, 255 \rangle;$   
 $4, 5: \langle 8, 16, 190, 429, 433 \rangle;$   
 $5, 2: \langle 363, 367 \rangle;$   
 $7, 3: \langle 13, 23, 191 \rangle; \dots \rangle$

be, 178239:

$\langle 1, 2: \langle 17, 25 \rangle;$   
 $4, 5: \langle 17, 191, 291, 430, 434 \rangle;$   
 $5, 3: \langle 14, 19, 101 \rangle; \dots \rangle$



# Pozicioni indeksi

- Pozicioni indeksi su dobra zamena za dvorečne indekse
- Liste pojava u **nepozicionom** indeksu: svaka pojava je docID
- Liste pojava u **pozicionom** indeksu: svaka pojava je docID i **lista pozicija**
- Primer:  $to_1$   $be_2$   $or_3$   $not_4$   $to_5$   $be_6$

to, 993427:

$\langle 1, 6: \langle 7, 18, 33, 72, 86, 231 \rangle;$   
 $2, 5: \langle 1, 17, 74, 222, 255 \rangle;$   
 $4, 5: \langle 8, 16, 190, 429, 433 \rangle;$   
 $5, 2: \langle 363, 367 \rangle;$   
 $7, 3: \langle 13, 23, 191 \rangle; \dots \rangle$

be, 178239:

$\langle 1, 2: \langle 17, 25 \rangle;$   
 $4, 5: \langle 17, 191, 291, 430, 434 \rangle;$   
 $5, 3: \langle 14, 19, 101 \rangle; \dots \rangle$

# Pozicioni indeksi

- Pozicioni indeksi su dobra zamena za dvorečne indekse
- Liste pojava u **nepozicionom** indeksu: svaka pojava je docID
- Liste pojava u **pozicionom** indeksu: svaka pojava je docID i **lista pozicija**
- Primer:  $to_1$   $be_2$   $or_3$   $not_4$   $to_5$   $be_6$

to, 993427:

$\langle 1, 6: \langle 7, 18, 33, 72, 86, 231 \rangle;$   
 $2, 5: \langle 1, 17, 74, 222, 255 \rangle;$   
 $4, 5: \langle 8, 16, 190, 429, 433 \rangle;$   
 $5, 2: \langle 363, 367 \rangle;$   
 $7, 3: \langle 13, 23, 191 \rangle; \dots \rangle$

be, 178239:

$\langle 1, 2: \langle 17, 25 \rangle;$   
 $4, 5: \langle 17, 191, 291, 430, 434 \rangle;$   
 $5, 3: \langle 14, 19, 101 \rangle; \dots \rangle$

# Pozicioni indeksi

- Pozicioni indeksi su dobra zamena za dvorečne indekse
- Liste pojava u **nepozicionom** indeksu: svaka pojava je docID
- Liste pojava u **pozicionom** indeksu: svaka pojava je docID i **lista pozicija**
- Primer:  $to_1$   $be_2$   $or_3$   $not_4$   $to_5$   $be_6$

to, 993427:

$\langle 1, 6: \langle 7, 18, 33, 72, 86, 231 \rangle;$   
 $2, 5: \langle 1, 17, 74, 222, 255 \rangle;$   
 $4, 5: \langle 8, 16, 190, 429, 433 \rangle;$   
 $5, 2: \langle 363, 367 \rangle;$   
 $7, 3: \langle 13, 23, 191 \rangle; \dots \rangle$

be, 178239:

$\langle 1, 2: \langle 17, 25 \rangle;$   
 $4, 5: \langle 17, 191, 291, 430, 434 \rangle;$   
 $5, 3: \langle 14, 19, 101 \rangle; \dots \rangle$

# Pozicioni indeksi

- Pozicioni indeksi su dobra zamena za dvorečne indekse
- Liste pojava u **nepozicionom** indeksu: svaka pojava je docID
- Liste pojava u **pozicionom** indeksu: svaka pojava je docID i **lista pozicija**
- Primer:  $to_1$   $be_2$   $or_3$   $not_4$   $to_5$   $be_6$

to, 993427:

$\langle 1, 6: \langle 7, 18, 33, 72, 86, 231 \rangle;$   
 $2, 5: \langle 1, 17, 74, 222, 255 \rangle;$   
 $4, 5: \langle 8, 16, 190, 429, 433 \rangle;$   
 $5, 2: \langle 363, 367 \rangle;$   
 $7, 3: \langle 13, 23, 191 \rangle; \dots \rangle$

be, 178239:

$\langle 1, 2: \langle 17, 25 \rangle;$   
 $4, 5: \langle 17, 191, 291, 430, 434 \rangle;$   
 $5, 3: \langle 14, 19, 101 \rangle; \dots \rangle$

# Pozicioni indeksi

- Pozicioni indeksi su dobra zamena za dvorečne indekse
- Liste pojava u **nepozicionom** indeksu: svaka pojava je docID
- Liste pojava u **pozicionom** indeksu: svaka pojava je docID i **lista pozicija**
- Primer:  $to_1$   $be_2$   $or_3$   $not_4$   $to_5$   $be_6$

to, 993427:

$\langle 1, 6: \langle 7, 18, 33, 72, 86, 231 \rangle;$   
 $2, 5: \langle 1, 17, 74, 222, 255 \rangle;$   
 $4, 5: \langle 8, 16, 190, 429, 433 \rangle;$   
 $5, 2: \langle 363, 367 \rangle;$   
 $7, 3: \langle 13, 23, 191 \rangle; \dots \rangle$

be, 178239:

$\langle 1, 2: \langle 17, 25 \rangle;$   
 $4, 5: \langle 17, 191, 291, 430, 434 \rangle;$   
 $5, 3: \langle 14, 19, 101 \rangle; \dots \rangle$

# Pozicioni indeksi

- Pozicioni indeksi su dobra zamena za dvorečne indekse
- Liste pojava u **nepozicionom** indeksu: svaka pojava je docID
- Liste pojava u **pozicionom** indeksu: svaka pojava je docID i **lista pozicija**
- Primer:  $to_1$   $be_2$   $or_3$   $not_4$   $to_5$   $be_6$

to, 993427:

$\langle 1, 6: \langle 7, 18, 33, 72, 86, 231 \rangle;$   
 $2, 5: \langle 1, 17, 74, 222, 255 \rangle;$   
 $4, 5: \langle 8, 16, 190, 429, 433 \rangle;$   
 $5, 2: \langle 363, 367 \rangle;$   
 $7, 3: \langle 13, 23, 191 \rangle; \dots \rangle$

be, 178239:

$\langle 1, 2: \langle 17, 25 \rangle;$   
 $4, 5: \langle 17, 191, 291, 430, 434 \rangle;$   
 $5, 3: \langle 14, 19, 101 \rangle; \dots \rangle$

# Pozicioni indeksi

- Pozicioni indeksi su dobra zamena za dvorečne indekse
- Liste pojava u **nepozicionom** indeksu: svaka pojava je docID
- Liste pojava u **pozicionom** indeksu: svaka pojava je docID i **lista pozicija**
- Primer:  $to_1$   $be_2$   $or_3$   $not_4$   $to_5$   $be_6$

to, 993427:

$\langle 1, 6: \langle 7, 18, 33, 72, 86, 231 \rangle;$   
 $2, 5: \langle 1, 17, 74, 222, 255 \rangle;$   
 $4, 5: \langle 8, 16, 190, 429, 433 \rangle;$   
 $5, 2: \langle 363, 367 \rangle;$   
 $7, 3: \langle 13, 23, 191 \rangle; \dots \rangle$

be, 178239:

$\langle 1, 2: \langle 17, 25 \rangle;$   
 $4, 5: \langle 17, 191, 291, 430, 434 \rangle;$   
 $5, 3: \langle 14, 19, 101 \rangle; \dots \rangle$

# Pozicioni indeksi

- Pozicioni indeksi su dobra zamena za dvorečne indekse
- Liste pojava u **nepozicionom** indeksu: svaka pojava je docID
- Liste pojava u **pozicionom** indeksu: svaka pojava je docID i **lista pozicija**
- Primer:  $to_1$   $be_2$   $or_3$   $not_4$   $to_5$   $be_6$

to, 993427:

$\langle 1, 6: \langle 7, 18, 33, 72, 86, 231 \rangle;$   
 $2, 5: \langle 1, 17, 74, 222, 255 \rangle;$   
 $4, 5: \langle 8, 16, 190, 429, 433 \rangle;$   
 $5, 2: \langle 363, 367 \rangle;$   
 $7, 3: \langle 13, 23, 191 \rangle; \dots \rangle$

be, 178239:

$\langle 1, 2: \langle 17, 25 \rangle;$   
 $4, 5: \langle 17, 191, 291, 430, 434 \rangle;$   
 $5, 3: \langle 14, 19, 101 \rangle; \dots \rangle$



# Pozicioni indeksi

- Pozicioni indeksi su dobra zamena za dvorečne indekse
- Liste pojava u **nepozicionom** indeksu: svaka pojava je docID
- Liste pojava u **pozicionom** indeksu: svaka pojava je docID i **lista pozicija**
- Primer:  $to_1$   $be_2$   $or_3$   $not_4$   $to_5$   $be_6$

to, 993427:

$\langle 1, 6: \langle 7, 18, 33, 72, 86, 231 \rangle;$   
 $2, 5: \langle 1, 17, 74, 222, 255 \rangle;$   
 $4, 5: \langle 8, 16, 190, 429, 433 \rangle;$   
 $5, 2: \langle 363, 367 \rangle;$   
 $7, 3: \langle 13, 23, 191 \rangle; \dots \rangle$

be, 178239:

$\langle 1, 2: \langle 17, 25 \rangle;$   
 $4, 5: \langle 17, 191, 291, 430, 434 \rangle;$   
 $5, 3: \langle 14, 19, 101 \rangle; \dots \rangle$

# Pozicioni indeksi

- Pozicioni indeksi su dobra zamena za dvorečne indekse
- Liste pojava u **nepozicionom** indeksu: svaka pojava je docID
- Liste pojava u **pozicionom** indeksu: svaka pojava je docID i **lista pozicija**
- Primer:  $to_1$   $be_2$   $or_3$   $not_4$   $to_5$   $be_6$

to, 993427:

$\langle 1, 6: \langle 7, 18, 33, 72, 86, 231 \rangle;$   
 $2, 5: \langle 1, 17, 74, 222, 255 \rangle;$   
 $4, 5: \langle 8, 16, 190, 429, 433 \rangle;$   
 $5, 2: \langle 363, 367 \rangle;$   
 $7, 3: \langle 13, 23, 191 \rangle; \dots \rangle$

be, 178239:

$\langle 1, 2: \langle 17, 25 \rangle;$   
 $4, 5: \langle 17, 191, 291, 430, 434 \rangle;$   
 $5, 3: \langle 14, 19, 101 \rangle; \dots \rangle$

# Pozicioni indeksi

- Pozicioni indeksi su dobra zamena za dvorečne indekse
- Liste pojava u **nepozicionom** indeksu: svaka pojava je docID
- Liste pojava u **pozicionom** indeksu: svaka pojava je docID i **lista pozicija**
- Primer:  $to_1$   $be_2$   $or_3$   $not_4$   $to_5$   $be_6$

to, 993427:

$\langle 1, 6: \langle 7, 18, 33, 72, 86, 231 \rangle;$   
 $2, 5: \langle 1, 17, 74, 222, 255 \rangle;$   
 $4, 5: \langle 8, 16, 190, 429, 433 \rangle;$   
 $5, 2: \langle 363, 367 \rangle;$   
 $7, 3: \langle 13, 23, 191 \rangle; \dots \rangle$

be, 178239:

$\langle 1, 2: \langle 17, 25 \rangle;$   
 $4, 5: \langle 17, 191, 291, 430, 434 \rangle;$   
 $5, 3: \langle 14, 19, 101 \rangle; \dots \rangle$

# Pozicioni indeksi

- Pozicioni indeksi su dobra zamena za dvorečne indekse
- Liste pojava u **nepozicionom** indeksu: svaka pojava je docID
- Liste pojava u **pozicionom** indeksu: svaka pojava je docID i **lista pozicija**
- Primer:  $to_1$   $be_2$   $or_3$   $not_4$   $to_5$   $be_6$

to, 993427:

$\langle 1, 6: \langle 7, 18, 33, 72, 86, 231 \rangle;$   
 $2, 5: \langle 1, 17, 74, 222, 255 \rangle;$   
 $4, 5: \langle 8, 16, 190, 429, 433 \rangle;$   
 $5, 2: \langle 363, 367 \rangle;$   
 $7, 3: \langle 13, 23, 191 \rangle; \dots \rangle$

be, 178239:

$\langle 1, 2: \langle 17, 25 \rangle;$   
 $4, 5: \langle 17, 191, 291, 430, 434 \rangle;$   
 $5, 3: \langle 14, 19, 101 \rangle; \dots \rangle$

Document 4 je pogodak!

# Vežba

Prikazan je deo pozicionog indeksa u formatu: term: doc1: ⟨position1, position2, ...⟩; doc2: ⟨position1, position2, ...⟩; etc.

*angels*: 2: ⟨36,174,252,651⟩; 4: ⟨12,22,102,432⟩; 7: ⟨17⟩;  
*fools*: 2: ⟨1,17,74,222⟩; 4: ⟨8,78,108,458⟩; 7: ⟨3,13,23,193⟩;  
*fear*: 2: ⟨87,704,722,901⟩; 4: ⟨13,43,113,433⟩; 7: ⟨18,328,528⟩;  
*in*: 2: ⟨3,37,76,444,851⟩; 4: ⟨10,20,110,470,500⟩; 7: ⟨5,15,25,195⟩;  
*rush*: 2: ⟨2,66,194,321,702⟩; 4: ⟨9,69,149,429,569⟩; 7: ⟨4,14,404⟩;  
*to*: 2: ⟨47,86,234,999⟩; 4: ⟨14,24,774,944⟩; 7: ⟨199,319,599,709⟩;  
*tread*: 2: ⟨57,94,333⟩; 4: ⟨15,35,155⟩; 7: ⟨20,320⟩;  
*where*: 2: ⟨67,124,393,1001⟩; 4: ⟨11,41,101,421,431⟩; 7: ⟨16,36,736⟩;

Koji dokument(i) zadovoljava sledeće upite-fraze: “fools rush in”, “fools rush in” AND “angels fear to tread”

# Blizinska pretraga

- Pozicioni indeks se može koristiti i za blizinsku pretragu
- Na primer imamo upit **pretraga /3 metapodatak**
- Pronađi sve dokumente koji sadrže pretraga i metapodatak na rastojanju od najviše tri reči
- Pretraga po metapodacima može biti implementirana upotrebom Lucene biblioteke je pogodak
- Rezultati pretraga digitalne biblioteke su prikazani u MARC 21 formatu metapodataka nije pogodak

# Blizinska pretraga

- Najjednostavniji algoritam: Dekartov proizvod pozicija za (i) employment i (ii) place

# Blizinska pretraga

- Najjednostavniji algoritam: Dekartov proizvod pozicija za (i) employment i (ii) place
- Neefikasan za česte reči, naročito za stop reči



# Blizinska pretraga

- Najjednostavniji algoritam: Dekartov proizvod pozicija za (i) employment i (ii) place
- Neefikasan za česte reči, naročito za stop reči
- Hoćemo da vratimo pozicije, ne samo listu dokumenata

# Blizinska pretraga

- Najjednostavniji algoritam: Dekartov proizvod pozicija za (i) employment i (ii) place
- Neefikasan za česte reči, naročito za stop reči
- Hoćemo da vratimo pozicije, ne samo listu dokumenata
- Ovo je važno za dinamičke sažetke i sl.

## Blizinski presek

```

PositionalIntersect( $p_1, p_2, k$ )
1   $answer \leftarrow \langle \rangle$ 
2  while  $p_1 \neq \text{nil}$  and  $p_2 \neq \text{nil}$ 
3  do if  $\text{docID}(p_1) = \text{docID}(p_2)$ 
4      then  $I \leftarrow \langle \rangle$ 
5           $pp_1 \leftarrow \text{positions}(p_1)$ 
6           $pp_2 \leftarrow \text{positions}(p_2)$ 
7          while  $pp_1 \neq \text{nil}$ 
8              do while  $pp_2 \neq \text{nil}$ 
9                  do if  $|\text{pos}(pp_1) - \text{pos}(pp_2)| \leq k$ 
10                     then  $\text{Add}(I, \text{pos}(pp_2))$ 
11                     else if  $\text{pos}(pp_2) > \text{pos}(pp_1)$ 
12                         then break
13                      $pp_2 \leftarrow \text{next}(pp_2)$ 
14                     while  $I \neq \langle \rangle$  and  $|I[0] - \text{pos}(pp_1)| > k$ 
15                         do  $\text{Delete}(I[0])$ 
16                     for each  $ps \in I$ 
17                         do  $\text{Add}(answer, \langle \text{docID}(p_1), \text{pos}(pp_1), ps \rangle)$ 
18                      $pp_1 \leftarrow \text{next}(pp_1)$ 
19                  $p_1 \leftarrow \text{next}(p_1)$ 
20                  $p_2 \leftarrow \text{next}(p_2)$ 
21             else if  $\text{docID}(p_1) < \text{docID}(p_2)$ 
22                 then  $p_1 \leftarrow \text{next}(p_1)$ 
23             else  $p_2 \leftarrow \text{next}(p_2)$ 
24  return  $answer$ 

```

# Kombinovanje indeksa

- Dvorečni i pozicioni indeksi se mogu uspešno kombinovati
- Mnogo dvoreči su jako česte: Michael Jackson, Britney Spears, itd.
- Za ove dvoreči ubrzanje u odnosu na pozicioni indeks je značajno
- Način za kombinovanje: uključi česte dvoreči kao termine u rečnik; ostale fraze računaj pomoću pozicionog indeksa
- Williams et al. (2004): još bolja kombinovana pretraga, brža od pozicionog indeksa uz 26% više prostora za indeks, pored rednog broja dokumenta u kojem se nalazi term i liste pozicija skladišti i naredni term u dokumentu

# „Pozicioni“ upiti i Google

- Za web pretraživače pozicioni upiti su bili mnogo skuplji od običnih Bulovih upita
- *“stanford university palo alto”*
- Zašto?
- Može li se to pokazati?

# “Pozicioni” upiti i Google

