Recuperação Booleana

Prof. Dr. Leandro Balby Marinho

http//www.dsc.ufcg.edu.br/~lbmarinho

UFCG CEEI Departamento de Sistemas e Computação

Sistemas de Recuperação da Informação

(Slides Adaptados de Cristopher D. Manning)

Roteiro

1) Definição
2) Matriz Binária Termo-Documento
3) Indíce Invertido
4) Indexação
5) Otimização de Consultas
6) Modelo Booleano

Definição

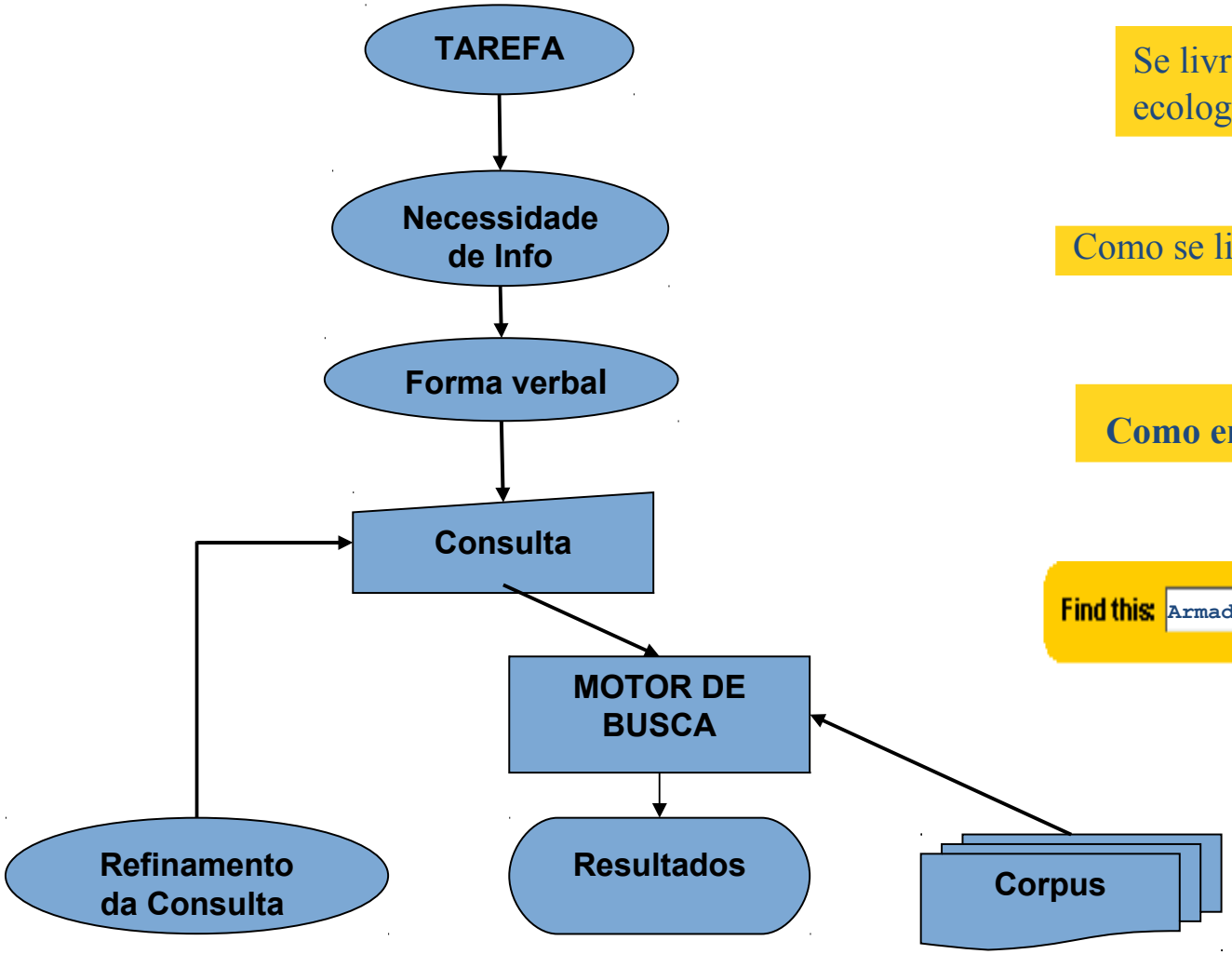
"Recuperação da Informação se preocupa em encontrar objetos (normalmente documentos) de natureza desestruturada (tipicamente texto) que satisfazem necessidades de informação em grandes coleções de documentos (geralmente armazenados em computador)." (Manning et al. 2009)

"Recuperação da Informação se preocupa em representar, buscar e manipular grandes coleções de texto e outros dados da linguagem Humana." (Büttcher et al. 2010)

Conceitos-chave

- Documentos: entidade contendo informações.
- Documentos desestruturados: ausência de estrutura e semântica clara processáveis por computador.
- Necessidade de informação: tópico sobre o qual o usuário quer saber.
- Consulta: forma como o usuário transmite sua necessidade de informação ao sistema.

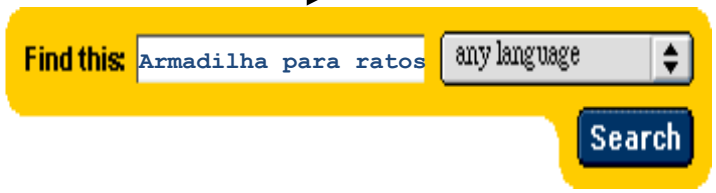
Modelo Clássico de Busca



Se livrar de ratos de forma ecologicamente correta

Como se livrar de ratos sem matá-los

Como encurralar ratos vivos?



Exemplo

- Quais peças de Shakespeare contém ***Brutus*** *AND* ***César*** mas *NOT* ***Calpurnia***?
- Poderia-se procurar em todas as peças de Shakespeare contendo ***Brutus*** AND ***César,*** e excluindo as peças contendo ***Calpurnia***

- Por que não é uma boa solução?
  - Inviável para grandes coleções de documentos
  - Pode-se querer outras operações (e.g., achar a palavra r***omanos*** perto de ***compatriota***) não é viável
  - Como lidar com ambiguidade?
  - Pode-se querer a ordenação dos resultados

Matriz Binária Termo-Documento

	Antônio e Cleópatra	Júlio César	A Tempestade	Hamlet	Othello	Macbeth
Antônio	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
César	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleópatra	1	0	0	0	0	0
misericórdia	1	0	1	1	1	1
pior	1	0	1	1	1	0

Brutus *AND* ***César*** *mas NOT* ***Calpurnia***

1 se peça contém termo, 0 de outra forma

Vetores de Incidência

- Um vetor 0/1 representa cada termo
- Para responder a consulta: *AND booleano entre* os vetores para ***Brutus, César*** e ***Calpurnia*** (complemento)

```
      110100
AND   110111
      101111
      ------
      100100
```

Qualidade dos Resultados

- Um documento é relevante na medida que ele satisfaz a necessidade de informação do usuário.
- De forma a avaliar a efetividade de um sistema de RI usa-se normalmente duas estatísticas-chave:

- Precisão (Precision): Que fração dos resultados recuperados são relevantes para a necessidade de informação?

- Cobertura (Recall): Que fração dos documentos relevantes foram recuperados?

Grandes Coleções

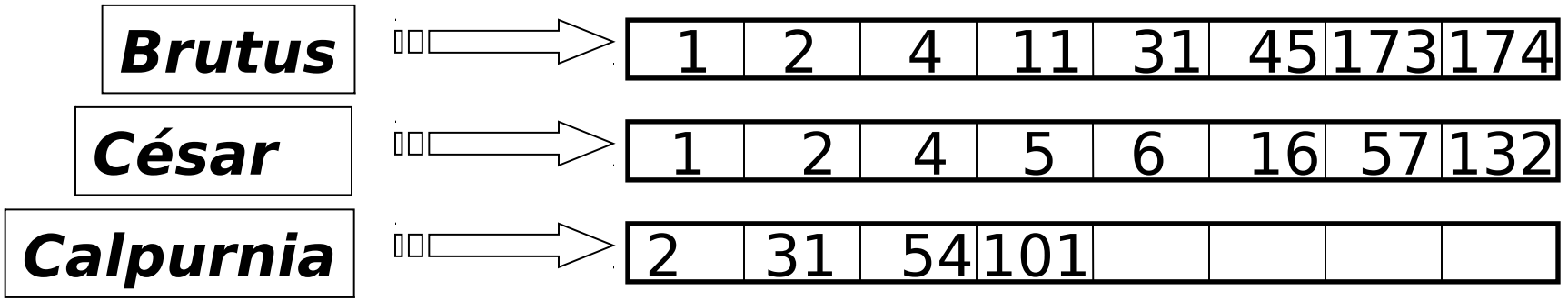
- Considere N = 1 milhão de documentos, cada um contento 1000 palavras.
- Considere 6 bytes/palavra incluindo espaço/pontuação
  - 6GB de dados nos documentos.
- Suponha M = 500K temos *distintos*.

Dispersão Termos/Documentos

- A matriz 500K x 1M tem meio trilhão de 0’s e 1’s.
- Mas não tem mais de um bilhão de 1’s.
  - A matriz é extremamente esparsa.
- Qual seria uma representação melhor?
  - Guardar apenas a posição dos 1.

Índice Invertido

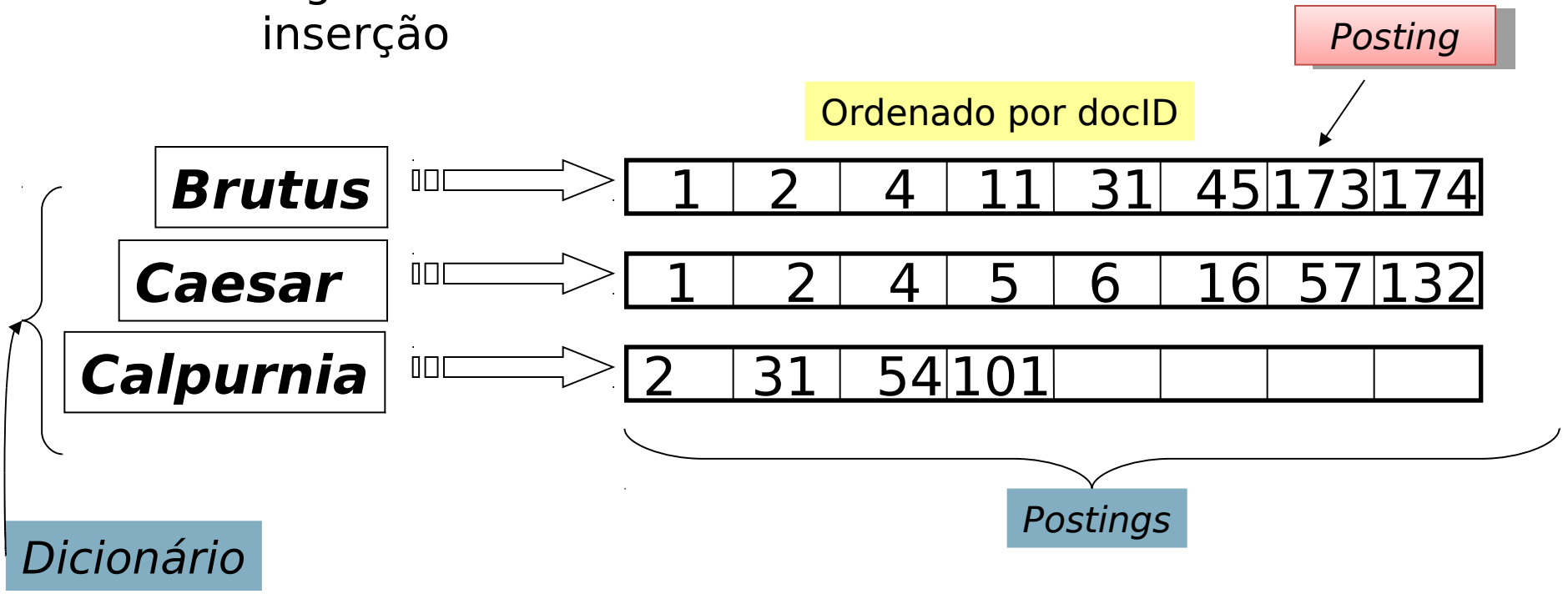
- Para cada termo *t*, devemos armazenar uma lista de todos os documentos contendo *t*.
  - Identificar cada documento um por um **docID**, um número serial
- Podemos usar arrays de tamanho fixo?



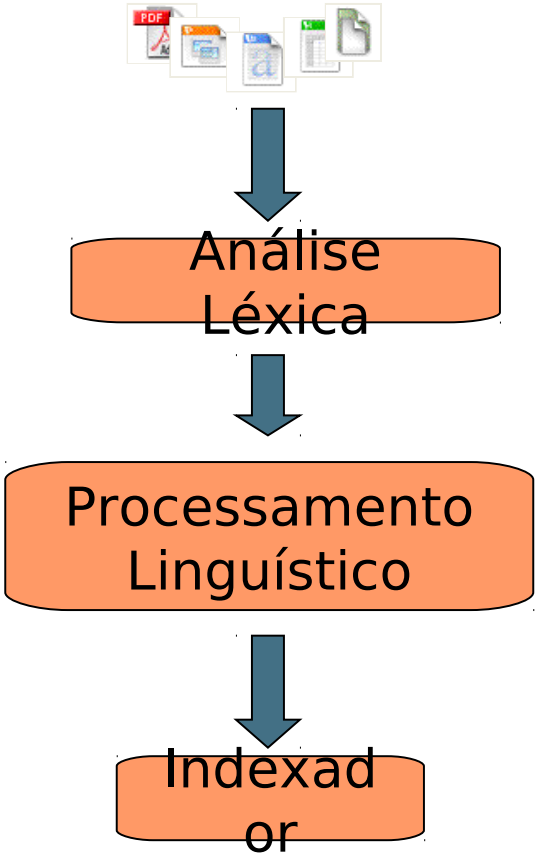
O que acontece se ***César*** é adicionado ao documento 14?

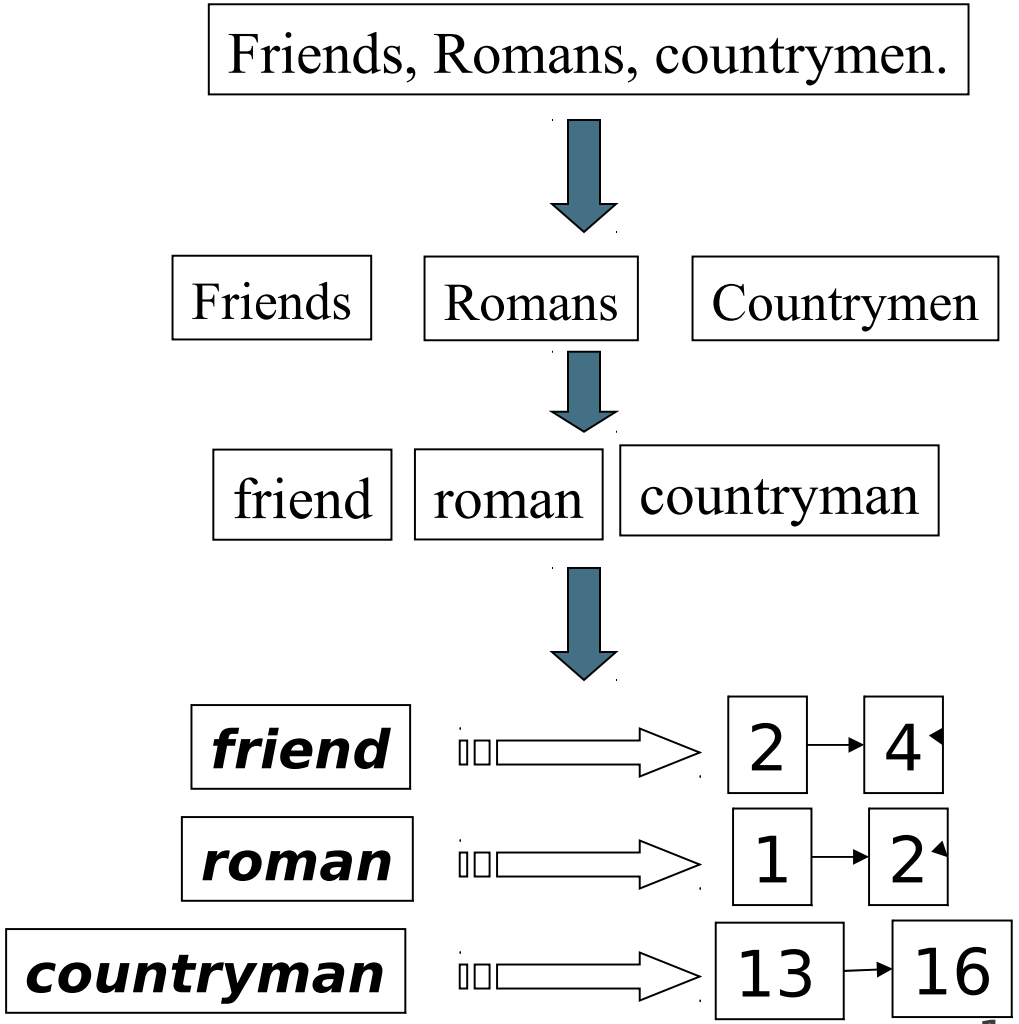
Estrututura de Dados

- Precisa-se de listas de tamanho variado
  - No disco, disposição contígua dos postings é melhor
  - Na memória, pode-se usar listas ligadas ou arrays de tamanho variável
    - Alguns tradeoffs entre tamanho/facilidade de inserção



Construção de Índice Invertido





Etapas da Indexação: Sequência de Tokens

- Sequência de pares (Tokens modificados, ID dos documentos)

Doc 1

I did enact Julius Caesar I was killed i' the Capitol; Brutus killed me.

Doc 2

So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious

Term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2

Etapas da Indexação: Ordenação

- Ordena por termo
  - E depois por docID

Etapa central de indexação

Term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

Term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

Etapas da Indexação: Dicionário e Postings

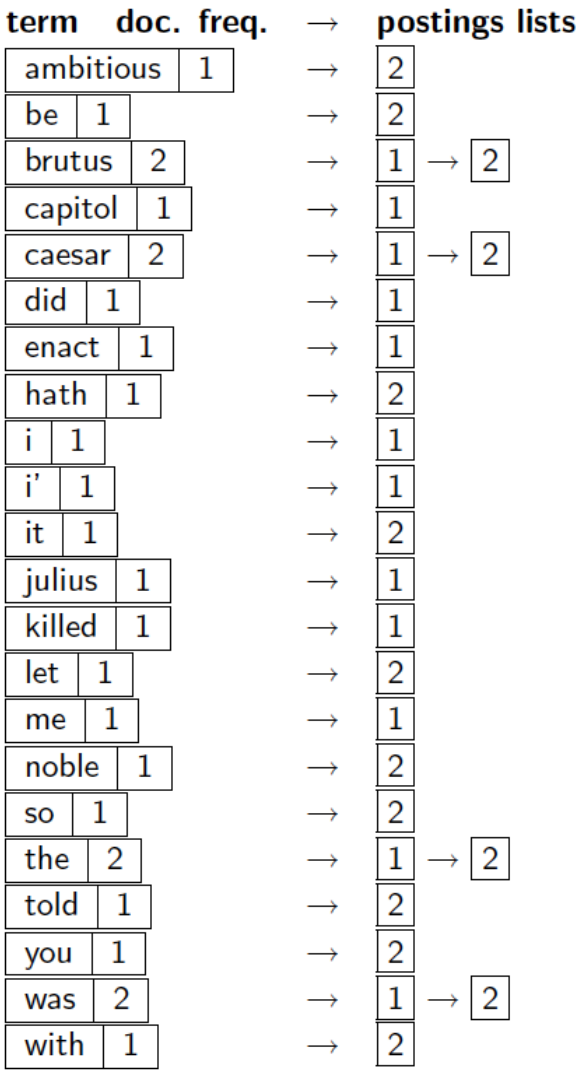
- Múltiplas entradas de termos no documento são combinadas.
- Dividir em dicionário e postings.
- Frequência dos documentos é adicionada.

Por que freq.?
Discutido depois

Term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

term	doc. freq.	→	postings lists
ambitious	1	→	2
be	1	→	2
brutus	2	→	1 → 2
capitol	1	→	1
caesar	2	→	1 → 2
did	1	→	1
enact	1	→	1
hath	1	→	2
i	1	→	1
i'	1	→	1
it	1	→	2
julius	1	→	1
killed	1	→	1
let	1	→	2
me	1	→	1
noble	1	→	2
so	1	→	2
the	2	→	1 → 2
told	1	→	2
you	1	→	2
was	2	→	1 → 2
with	1	→	2

Questões de Armazenamento



DocIds Ordenados

Termos e Freq.

Ponteiros

Mais tarde no curso:

- Como indexar de forma eficiente?
- Quanto armazenamento precisamos?

Exercício Livro Texto

- Exercício 1.2. Considere os seguintes documentos:

Doc 1	breakthrough drug for schizophrenia
Doc 2	new schizophrenia drug
Doc 3	new approach for treatment of schizophrenia
Doc 4	new hopes for schizophrenia patients

- a. Escreva a matriz de incidência termo-documento para essa coleção de documentos.
- b. Escreva o índice invertido para essa coleção.

Processamento de Consultas

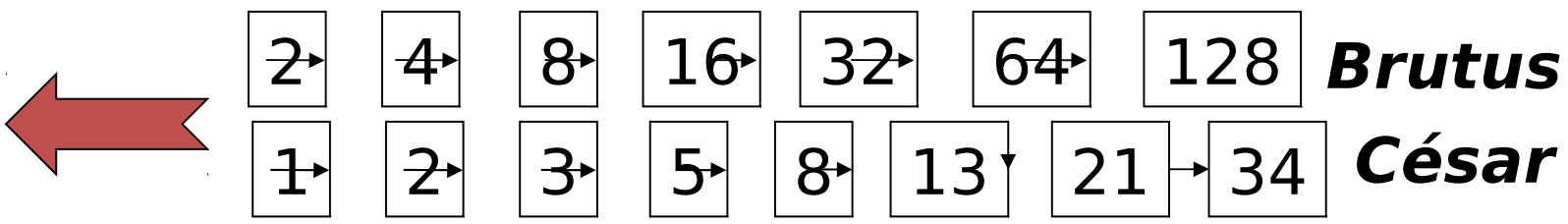
- Como processamos uma consulta?
  - Mais tarde – que tipos de consulta podem ser processadas?

Processamento de Consulta: AND

- Considere o processamento da consulta:

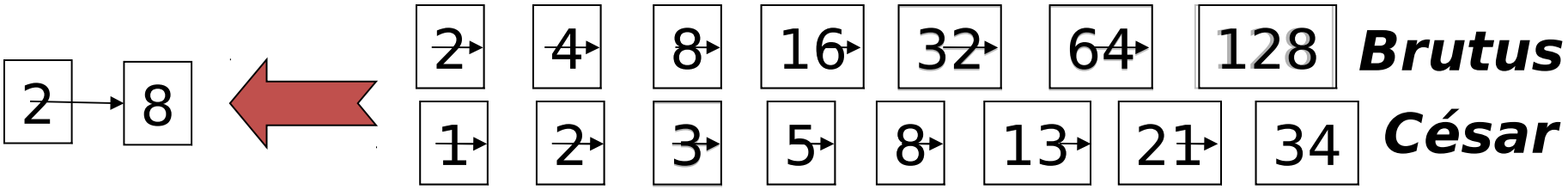
  Brutus *AND* ***César***

  - Ache ***Brutus*** no Dicionário;
    - Recupere seus postings.
  - Ache ***César*** no Dicionário;
    - Recupere seus postings.
  - Faça um “Merge” nos dois postings:



O Merge

- “Atravesse” as duas listas de postings simultaneamente, em tempo linear ao número total de entradas dos postings



Se os tamanhos da lista são *x* e *y*, o merge realiza O(*x*+*y*) operações.

Crucial: postings são ordenadas por docID.

Um algoritmo Merge

```
INTERSECT(p1, p2)
 1  answer ← ⟨ ⟩
 2  while p1 ≠ NIL and p2 ≠ NIL
 3  do if docID(p1) = docID(p2)
 4        then ADD(answer, docID(p1))
 5             p1 ← next(p1)
 6             p2 ← next(p2)
 7        else if docID(p1) < docID(p2)
 8                then p1 ← next(p1)
 9                else p2 ← next(p2)
10  return answer
```

Consultas Booleanas: Casamento Exato

- O modelo de recuperação Booleano é capaz de responder uma consulta Booleana:
  - Consultas Booleanas usam os operadores *AND, OR* e *NOT* para combinar termos de consulta
    - Representa cada documento como um conjunto de palavras
    - Preciso: documento casa com a consulta ou não.
  - Modelo mais simplista para basear um sistema de RI
- Principal ferramenta comercial de RI por três décadas (e.g. www.westlaw.com).
- Muitos sistemas de busca que você usa ainda usam o modelo Boolean:
  - Email, Mac OS X Spotlight

Consultas Booleanas Gerais

- Exercício: Adapte o merge para as seguintes consultas:

  Brutus *AND NOT* ***César***

  Brutus *OR NOT* ***César***

Ainda podemos rodar o merge em tempo linear O(x+y)?

O que se pode atingir?

Merging

O que dizer a respeito de consultas Booleanas arbitrárias?

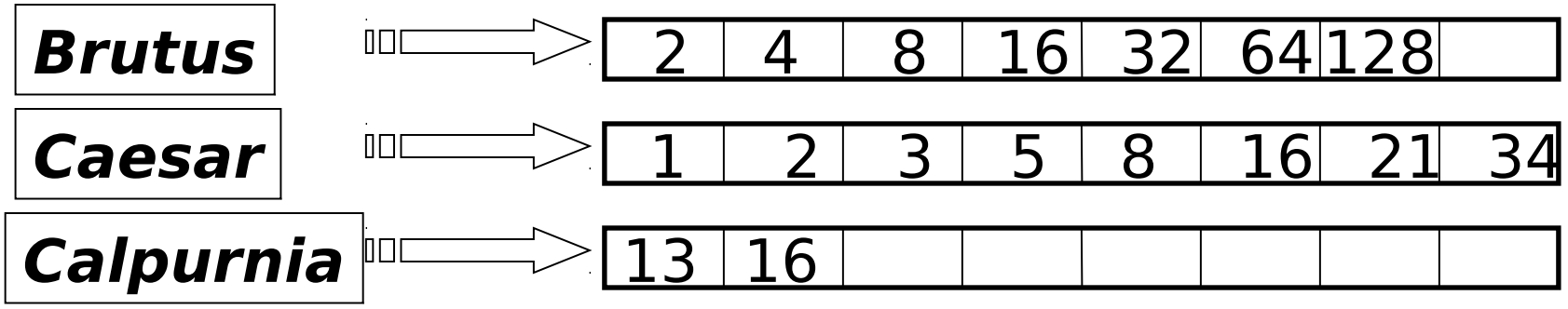
***(Brutus** OR **César)** AND NOT*

(Antony** OR **Cleopatra)

- Podemos sempre rodar o merge em tempo “linear”?
  - Linear em que?
- Podemos fazer melhor?

Otimização de Consultas

- Qual a melhor ordem para o processamento de consultas?
- Considere uma consulta que é um *AND* de *n* termos.
- Para cada um dos *n* termos, obtenha seus postings, e depois faça um *AND* neles.

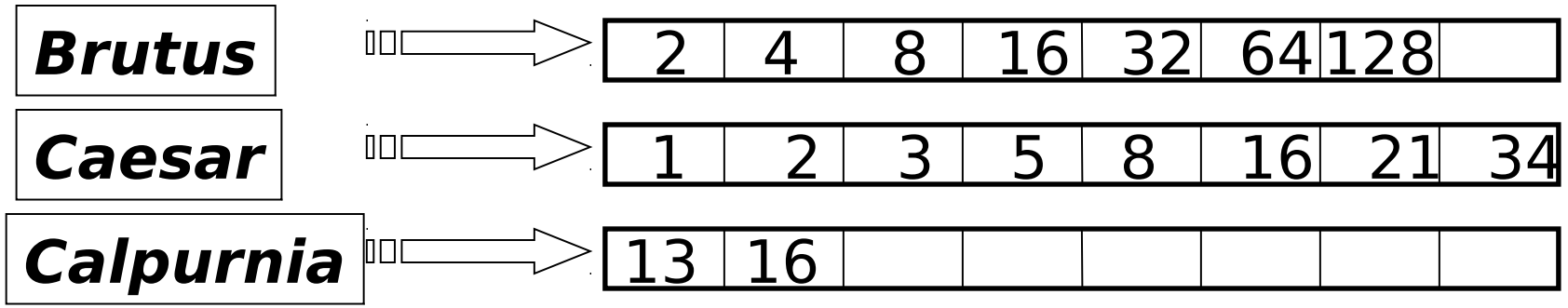


Consulta: ***Brutus*** *AND* ***Calpurnia*** *AND* ***César***

Exemplo de Otimização de Consulta

- Processar na ordem crescente de freq.

Por isso é importante guardar a freq. dos docs. na memória



Execute a consulta como (***Calpurnia*** *AND* ***Brutus)*** *AND* ***César***

Otimização de Consulta Genérica

- e.g., *(**madding** OR **crowd**) AND (**ignoble** OR **strife**)*
- Obter a freq. de doc. para todos os termos.
- Estimar o tamanho de cada *OR* pela soma das freq. dos doc.
- Processar em ordem crescente dos tamanhos dos *OR*.

Exercício

- Sugira uma ordem de processamento de consulta para

***(tangerine** OR **trees)** AND*
***(marmalade** OR **skies)** AND*
(kaleidoscope** OR **eyes)

Termo	Freq
eyes	213312
kaleidoscope	87009
marmalade	107913
skies	271658
tangerine	46653
trees	316812

O que vem Adiante

- E quanto a frases?
  - ***Sistema Operacional***
- Proximidade: Ache ***Gates*** *NEAR* ***Microsoft***.
  - Índices que capturam a posição da informação nos documentos.
- Zonas em documentos: Ache docs. com (*autor* = ***Ullman***) *AND* (text contém ***automata***).

Acúmulo de Evidência

- 1 vs. 0 ocorrências de um termo de busca
  - 2 vs. 1 ocorrências
  - 3 vs. 2 ocorrências, etc.
  - Geralmente mais significa melhor
- Precisa-se das freq. de termos nos docs.

Busca baseada em Ranking

- Consultas Booleanas produzem a inclusão ou exclusão de docs.
- Frequentemente queremos ordenar os resultados
  - Precisa-se mensurar a proximidade da consulta com cada doc.
  - Precisa-se decidir se os docs apresentados ao usuário são documentos únicos ou um grupo de docs cobrindo vários aspectos da consulta.

Recursos da Aula de Hoje

- *Introduction to Information Retrieval*, capítulo 1.
- Information Retrieaval, Implementing and Evaluating Search Engines, capítulo 1.