# Übersicht

# Suffix Arrays

Gegeben ist ein String *abracadabra*$.
Seine Suffixe sind:
*abracadabra*
*bracadabra*
*racadabra*
*acadabra*
*cadabra*
*adabra*
*dabra*
*abra*
*bra*
*ra*
*a*

Wir können die Suffixe lexikographisch sortieren:

*a*
*abra*
*abracadabra*
*acadabra*
*adabra*
*bra*
*bracadabra*
*cadabra*
*dabra*
*ra*
*racadabra*

Wie lange dauert das? Wieviel Speicher brauchen wir?

*abracadabra* 0
*bracadabra* 1
*racadabra* 2
*acadabra* 3
*cadabra* 4
*adabra* 5
*dabra* 6
*abra* 7
*bra* 8
*ra* 9
*a* 10

Wir müssen nur die Anfangsindexe speichern.

*a* 10
*abra* 7
*abracadabra* 0
*acadabra* 3
*adabra* 5
*bra* 8
*bracadabra* 1
*cadabra* 4
*dabra* 6
*ra* 9
*racadabra* 2

Wir müssen nur die Anfangsindexe speichern:
Suffix Array: 10, 7, 0, 3, 5, 8, 1, 4, 6, 9, 2

*a* 10
*abra* 7
*abracadabra* 0
*acadabra* 3
*adabra* 5
*bra* 8
*bracadabra* 1
*cadabra* 4
*dabra* 6
*ra* 9
*racadabra* 2

Wir können ein Wort $u$ in $O(|u| \log n)$ Schritten suchen, falls wir ein Suffix Array der Größe $n$ haben.

# Konstruktion eines Suffix Arrays

- $O(n^2 \log n)$: Alle Suffixe sortieren
- $O(n^2)$ Alle Suffixe mit Radix-Sort sortieren
- $O(n \log n)$: Prefix Doubling
- $O(n)$: Rekursive Verfahren

# Prefix Doubling

Es sei $S_i^l$ die ersten $l$ Zeichen des Suffixes der an der $i$ten Position beginnt.

Oder: $S_i^l = w_i \dots w_{i+l-1}$

Falls $w$ zu kurz ist, dann endet $S_i^l$ früher.

Beispiel: $w = abracadabra$

$S_1^4 = abra$

$S_5^6 = cadabr$

$S_{10}^4 = ra$

# Prefix Doubling

Ziel: Finde einfache Labels $L_i^l$ mit

$$L_i^l < L_j^l \iff S_i^l < S_j^l.$$

Wenn wir alle $L_i^n$ haben, können wir sie anstatt der $S_i^n$ sortieren.

Was eignet sich für $L_i^1$?

Wir nehmen einfach $L_i^1 = w_i$.

# Prefix Doubling

Wenn wir alle $L_i^l$ haben, können wir $L_i^{2l}$ berechnen:
Setze zunächst $L_i^{2l} = (L_i^l, L_{i+l}^l)$.
Es gilt offensichtlich jetzt:

$$(L_i^l, L_{i+l}^l) < (L_j^l, L_{j+l}^l) \iff S_i^{2l} < S_j^{2l}$$

Jetzt können wir alle $(L_i^l, L_{i+l}^l)$ sortieren, zum Beispiel:

$(3,2), (3,2), (3,4), (5,2), (5,2), (5,4), (6,7)$.

Dann können wir neue Labels $L_i^{2l}$ konstruieren:
$1, 1, 2, 3, 3, 4, 5$.

| $S_i^1$ | $L_i^1$ | $(L_i^1, L_{i+1}^1)$ | $i$ |
|---|---|---|---|
| a | 97 | $(97, 0)$ | 10 |
| abracadabra | 97 | $(97, 98)$ | 0 |
| abra | 97 | $(97, 98)$ | 7 |
| acadabra | 97 | $(97, 99)$ | 3 |
| adabra | 97 | $(97, 100)$ | 5 |
| bracadabra | 98 | $(98, 114)$ | 1 |
| bra | 98 | $(98, 114)$ | 8 |
| cadabra | 99 | $(99, 97)$ | 4 |
| dabra | 100 | $(100, 97)$ | 6 |
| racadabra | 114 | $(114, 97)$ | 2 |
| ra | 114 | $(114, 97)$ | 9 |

| $S_i^2$ | $L_i^2$ | $(L_i^2, L_{i+2}^2)$ | $i$ |
|---|---|---|---|
| a | 1 | $(1,0)$ | 10 |
| abracadabra | 2 | $(2,8)$ | 0 |
| abra | 2 | $(2,8)$ | 7 |
| acadabra | 3 | $(3,4)$ | 3 |
| adabra | 4 | $(4,2)$ | 5 |
| bra | 5 | $(5,1)$ | 8 |
| bracadabra | 5 | $(5,3)$ | 1 |
| cadabra | 6 | $(6,7)$ | 4 |
| dabra | 7 | $(7,5)$ | 6 |
| ra | 8 | $(8,0)$ | 9 |
| racadabra | 8 | $(8,6)$ | 2 |

| $S_i^4$ | $L_i^4$ | $(L_i^4, L_{i+4}^4)$ | $i$ |
|---|---|---|---|
| a | 1 | $(1,0)$ | 10 |
| abra | 2 | $(2,0)$ | 7 |
| abracadabra | 2 | $(2,7)$ | 0 |
| acadabra | 3 | $(3,2)$ | 3 |
| adabra | 4 | $(4,9)$ | 5 |
| bra | 5 | $(5,0)$ | 8 |
| bracadabra | 6 | $(6,4)$ | 1 |
| cadabra | 7 | $(7,5)$ | 4 |
| dabra | 8 | $(8,1)$ | 6 |
| ra | 9 | $(9,0)$ | 9 |
| racadabra | 10 | $(10,8)$ | 2 |

| $S_i^8$ | $L_i^8$ | $(L_i^8, L_{i+8}^8)$ | $i$ |
|---|---|---|---|
| a | 1 | $(1,0)$ | 10 |
| abra | 2 | $(2,0)$ | 7 |
| abracadabra | 3 | $(3,6)$ | 0 |
| acadabra | 4 | $(4,0)$ | 3 |
| adabra | 5 | $(5,0)$ | 5 |
| bra | 6 | $(6,0)$ | 8 |
| bracadabra | 7 | $(7,10)$ | 1 |
| cadabra | 8 | $(8,0)$ | 4 |
| dabra | 9 | $(9,0)$ | 6 |
| ra | 10 | $(10,0)$ | 9 |
| racadabra | 11 | $(11,1)$ | 2 |

| $S_i^1$ | $L_i^1$ | $(L_i^1, L_{i+1}^1)$ | $i$ |
|---|---|---|---|
| a | 97 | $(97, 0)$ | 10 |
| aaaaaaaaaaa | 97 | $(97, 97)$ | 0 |
| aaaaaaaaaa | 97 | $(97, 97)$ | 1 |
| aaaaaaaaa | 97 | $(97, 97)$ | 2 |
| aaaaaaaa | 97 | $(97, 97)$ | 3 |
| aaaaaaa | 97 | $(97, 97)$ | 4 |
| aaaaaa | 97 | $(97, 97)$ | 5 |
| aaaaa | 97 | $(97, 97)$ | 6 |
| aaaa | 97 | $(97, 97)$ | 7 |
| aaa | 97 | $(97, 97)$ | 8 |
| aa | 97 | $(97, 97)$ | 9 |

| $S_i^2$ | $L_i^2$ | $(L_i^2, L_{i+2}^2)$ | $i$ |
|---|---|---|---|
| a | 1 | $(1,0)$ | 10 |
| aa | 2 | $(2,0)$ | 9 |
| aaa | 2 | $(2,1)$ | 8 |
| aaaaaaaaaaa | 2 | $(2,2)$ | 0 |
| aaaaaaaaaa | 2 | $(2,2)$ | 1 |
| aaaaaaaaa | 2 | $(2,2)$ | 2 |
| aaaaaaaa | 2 | $(2,2)$ | 3 |
| aaaaaaa | 2 | $(2,2)$ | 4 |
| aaaaaa | 2 | $(2,2)$ | 5 |
| aaaaa | 2 | $(2,2)$ | 6 |
| aaaa | 2 | $(2,2)$ | 7 |

| $S_i^4$ | $L_i^4$ | $(L_i^4, L_{i+4}^4)$ | $i$ |
|---|---|---|---|
| a | 1 | $(1,0)$ | 10 |
| aa | 2 | $(2,0)$ | 9 |
| aaa | 3 | $(3,0)$ | 8 |
| aaaa | 4 | $(4,0)$ | 7 |
| aaaaa | 4 | $(4,1)$ | 6 |
| aaaaaa | 4 | $(4,2)$ | 5 |
| aaaaaaa | 4 | $(4,3)$ | 4 |
| aaaaaaaaaa | 4 | $(4,4)$ | 0 |
| aaaaaaaaa | 4 | $(4,4)$ | 1 |
| aaaaaaaa | 4 | $(4,4)$ | 2 |
| aaaaaaa | 4 | $(4,4)$ | 3 |

| $S_i^8$ | $L_i^8$ | $(L_i^8, L_{i+8}^8)$ | $i$ |
|---|---|---|---|
| a | 1 | $(1,0)$ | 10 |
| aa | 2 | $(2,0)$ | 9 |
| aaa | 3 | $(3,0)$ | 8 |
| aaaa | 4 | $(4,0)$ | 7 |
| aaaaa | 5 | $(5,0)$ | 6 |
| aaaaaa | 6 | $(6,0)$ | 5 |
| aaaaaaa | 7 | $(7,0)$ | 4 |
| aaaaaaaa | 8 | $(8,0)$ | 3 |
| aaaaaaaaa | 8 | $(8,1)$ | 2 |
| aaaaaaaaaa | 8 | $(8,2)$ | 1 |
| aaaaaaaaaaa | 8 | $(8,3)$ | 0 |

## Programm in Python

```
def suffix_array(s) :
  n = len(s); l = 1; labels = []
  for i in range(n) : labels.append(ord(s[i]))
  while l < n :
    pairlabels = []
    for i in range(n) :
      a, b = labels[i], 0
      if i + l < n : b = labels[i + l];
      pairlabels.append((a, b, i))
    pairlabels.sort()
    k, old_a, old_b = 0, −1, −1
    for a, b, i in pairlabels :
      if a ≠ old_a or b ≠ old_b : k, old_a, old_b = k + 1, a, b
      labels[i] = k
    l = l ∗ 2
  suffix_array = [0] ∗ n
  for i in range(n) : suffix_array[labels[i] − 1] = i
  return suffix_array
```