String Sheet ♫

https://github.com/koeppl/stringsheet

notation

- $\prec :=$ lexicographic order
- $\varepsilon :=$ empty string
- $T :=$ string, $T[|T|] :=$ \$, \$ $< T[i]\ \forall i \in [1..|T|)$
- $S \lhd T :\Leftrightarrow S[\mathrm{lcp}(S,T)] < T[\mathrm{lcp}(S,T)]$
- $S \prec T :\Leftrightarrow S \lhd T \vee S$ is proper prefix of T
- $w :=$ word size
- $n :=$ length of a given text T

alphabet Σ

- constant $:\Leftrightarrow |\Sigma| = \mathcal{O}(1)$
- integer $:\Leftrightarrow |\Sigma| = n^{\mathcal{O}(1)}$
- finite $:\Leftrightarrow |\Sigma| < \infty$
- constant $\subset$ integer $\subset$ finite

entropy for alphabet $\Sigma := \{c_1, \ldots, c_\sigma\}$:

- $H_0(T) := (1/|T|) \sum_{j=1}^{\sigma} n_j \lg(|T|/n_j)$, $n_j := |\{i : T[i] = c_j\}|$
- $H_k(T) := (1/|T|) \sum_{S \in \Sigma^k} |T_S| H_0(T_S)$, where T_S is the concatenation of each character in T that directly follows an occurrence of the substring $S \in \Sigma^k$ in T.

computational model

- cell probe model: no cost for computation, counts only memory access $\Rightarrow$ for lower bounds [20]
- pointer machine model: time for any random access is constant
- word RAM model: pointer machine model with access to $\lg n = \mathcal{O}(w)$ consecutive bits in constant time [12]
- Transdichotomous model: word RAM model with $\lg n = \Theta(w)$ [10]

Word RAM model supports word-packed string $T'[i] = T[(i-1)\lfloor w/\lg\sigma\rfloor + 1 \mathinner{..} (i+1)\lfloor w/\lg\sigma\rfloor]$ with constant time operations

arrays

- SA: $T[\mathsf{SA}[i-1] \mathinner{..} n] \prec T[\mathsf{SA}[i] \mathinner{..} n]\ \forall i \in [2 \mathinner{..} n]$, suffix array [17]
- $\mathsf{LCP}[i] = \mathrm{lcp}(\mathsf{SA}[i-1], \mathsf{SA}[i])\ \forall i \in [2 \mathinner{..} n]$, longest common prefix array
- $\mathsf{PLCP}[\mathsf{SA}[i]] := \mathsf{LCP}[i]$, permuted LCP array [15]
- $\mathsf{LPF}[j] := \max\{\ell \mid \exists i \in [1 \mathinner{..} j-1] : T[i \mathinner{..} i+\ell-1] = T[j \mathinner{..} j+\ell-1]\}$, longest previous factor array [9, 4]
- $\mathsf{C}[i] := |\{j \in [1 \mathinner{..} n] : T[j] < i\}|$

backward

- $\mathsf{LF}[i] := C[\mathsf{BWT}[i]] + \mathsf{BWT}.\mathrm{rank}_{L[i]}(i)$
- $\mathsf{LF}[i] = \mathsf{ISA}[\mathsf{SA}[i] - 1]$ [7]
- $\mathsf{BWT}[i] := T[\mathsf{SA}[i] - 1]\ \forall i \neq \mathsf{ISA}[1]$, $\mathsf{BWT}[\mathsf{ISA}[1]] := T[n] =$ \$, Burrows-Wheeler transform [2]
- $\Phi[i] := \mathsf{SA}[\mathsf{ISA}[i] - 1]\ \forall i \neq \mathsf{ISA}[1]$, $\Phi[\mathsf{ISA}[1]] := n$ [14]
- $\Phi^k[i] = \mathsf{SA}[(\mathsf{ISA}[i] + n - k - 1 \mod n) + 1]\ \forall i, k \in [1 \mathinner{..} n]$.

forward

- $\Psi[i] := \mathsf{ISA}[\mathsf{SA}[i] + 1]\ \forall i \neq \mathsf{ISA}[n]$
- $\Psi[\mathsf{ISA}[n]] := \mathsf{ISA}[1]$ [11]

order \ dir	$\leftarrow$	$\rightarrow$
SA	LF	Ψ
T	Φ	

- $\Psi^k[i] = \mathsf{SA}[(\mathsf{ISA}[i] + k - 1 \mod n) + 1]\ \forall i, k \in [1 \mathinner{..} n]$.
- $\Psi[i] = \mathsf{BWT}.\mathrm{select}_{T[\mathsf{SA}[i]]}(i - C[T[\mathsf{SA}[i]]])$ [16]

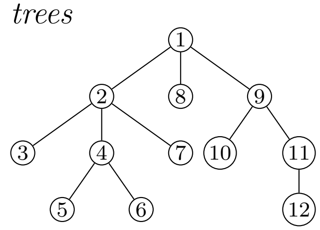
properties

- $\Psi\,|_{[C[c]+1, C[c+1]]}$ strictly increasing $\forall c \in \Sigma$.
- $T[\mathsf{SA}[i]] = c \wedge \mathsf{BWT}[\Psi[i]] = c\ \forall i \in [C[c]+1, C[c+1]]\ \forall c \in \Sigma$

reducible LCP values

- $\mathsf{LCP}[i] = \mathsf{PLCP}[\mathsf{SA}[i]]$ reducible $:\Leftrightarrow \mathsf{BWT}[i] = \mathsf{BWT}[i-1]$.
- $\mathsf{PLCP}[i+1]$ reducible $\Leftrightarrow T[i] = T[\Phi[i]] \wedge \Psi[\mathsf{ISA}[\Phi[i]]] = \Psi[\mathsf{ISA}[i]] - 1$. [18]
- $\mathsf{PLCP}[i]$ reducible $\Rightarrow \mathsf{PLCP}[i] = \mathsf{PLCP}[i-1] - 1 \wedge \Phi[i] = \Phi[i-1] + 1$. [14]

trees



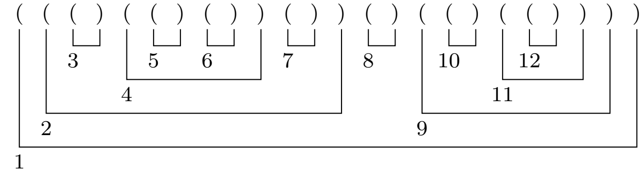
- DFUDS [1]
- LOUDS [13]
- BP [13]

$2n + \mathcal{O}(1)$ bits for n nodes.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
DFUDS	(	(	(	(	)	(	(	(	)	)	(	(	)	)	)	)	)	(	(	)	)	(	)	)
id		1				2				3	4			5	6	7	8	9			10	11		12

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
LOUDS	(	)	(	(	(	)	(	(	(	)	)	(	(	)	)	(	(	)	)	)	(	)	)	)	)
id	1		2	8	9		3	4	7			10	11			5	6				12				
parent			1				2				8	9			3	4			7	10	11		5	6	12

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
BP	(	(	(	)	(	(	)	(	)	)	(	)	)	(	)	(	(	)	(	(	)	)	)	)



regularity

- $p \leq |T|/2$ period of $T :\Leftrightarrow T[i+p] = T[i] \forall i \in [1 \mathinner{..} |T| - p]$
- T primitive $:\Leftrightarrow \nexists$ period of T
- exponent $\exp(T) := |T|/p$, p : minimal period of T

Periodicity Lemma [8]: T has periods p and p' with $p + p' \leq |T| \Rightarrow \gcd(p, p')$ is period of T.

- square U^2 irreducible $:\Leftrightarrow U$ is not a repetition.
- square U^2 minimal $:\Leftrightarrow \nexists P$ proper prefix of U^2 : P is square
- square U^2 minimal $\Rightarrow U^2$ is irreducible

Three Squares Lemma [5]: If

- U not a repetition,
- $\nexists j \in \mathbb{N} : V \neq U^j$,
- U^2 prefix of V^2, and
- V^2 proper prefix of W^2,

$\Rightarrow\ |W| \geq |U| + |V|$.

Lyndon

T Lyndon $:\Leftrightarrow T \lhd T[i\,..\,] \forall i \in [2\,..\,|T|]$
If T Lyndon, then

- $\nexists$ period of T
- T is border-free
- $T \prec S \wedge S$ Lyndon $\Rightarrow T \prec TS \prec S \Rightarrow TS$ Lyndon [3]

factorization

Let $T = F_1 \cdots F_z$.

Lyndon factorization

- F_i Lyndon $\forall i \in [1\,..\,z] \wedge F_i \succeq F_{i+1}\ \forall i \in [1\,..\,z)$ [3]
- $F_i = T[\mathsf{b}(F_i)\,..\,x]$ where $x \coloneqq \mathrm{argmin}_{j > \mathsf{b}(F_i)} \mathsf{ISA}[j] > \mathsf{ISA}[\mathsf{b}(F_i)]$
- F_i longest Lyndon prefix of $T[\mathsf{b}(F_i)\,..\,]$

Standard factorization: $T = UV$, where V is the smallest proper suffix of V.

Classic LZ77-factorization F_x is the shortest prefix of $F_x \cdots F_z$ occurring exactly once in $F_1 \cdots F_x$ [21].

LZ78 factorization $F_x = F'_x c$ with $F'_x = \mathrm{argmax}_{S \in \{F_y | y < x\} \cup \{\varepsilon\}} |S|$ and $c \in \Sigma$, $\forall x \in [1\,..\,z]$ [22].

s-factorization $F_x \coloneqq \mathrm{argmax}_{S \in \mathcal{S}_j(T) \cup \Sigma} |S|\ \forall x \in [1\,..\,z]$, where $j \coloneqq |F_1 \cdots F_{x-1}| + 1$ and $\mathcal{S}_j(T)$ is the set of substrings of T starting *strictly* before j [19].

Non-overlapping s-factorization
$F_x \coloneqq \mathrm{argmax}\{|S| \mid S \in \Sigma^*$ occurs in $T[1\,..\,j]$ or $S \in \Sigma\}\ \forall x \in [1\,..\,z]$, where $j \coloneqq |F_1 \cdots F_{x-1}|$.
SAIS

- $T[i] < T[i+1] \Rightarrow T[i\,..\,]$ is S-type ($T[i\,..\,] \prec T[i+1\,..\,]$)
- $T[i] > T[i+1] \Rightarrow T[i\,..\,]$ is L-type ($T[i\,..\,] \succ T[i+1\,..\,]$)
- $T[i] = T[i+1] \Rightarrow T[i\,..\,]$ has same type as $T[i+1\,..\,]$
- $T[i\,..\,]$ is S-type $\wedge\ T[i-1\,..\,]$ is L-type $\Rightarrow T[i\,..\,]$ is S^*-type

operations

- $T.\mathrm{psv}(j) \coloneqq \max\{k < j \mid T[k] < T[j]\}$ previous smaller value
- $T.\mathrm{nsv}(j) \coloneqq \min\{k > j \mid T[k] < T[j]\}$ next smaller value
- $T.\mathrm{rank}_c(j) \coloneqq |\{k \in [1\,..\,j] \mid T[k] = c\}|$ rank query
- $T.\mathrm{select}_c(k) \coloneqq \min\{j \mid T.\mathrm{rank}_c(j) = k\}$ select query
- $T.\mathrm{RMQ}[a,b] \coloneqq \mathrm{argmin}_{i \in [a\,..\,b]} T[i]$ range minimum query
- $T.\mathsf{lce}(a,b) \coloneqq \mathsf{lcp}(T[a\,..\,], T[b\,..\,]) = \mathsf{LCP}.\mathsf{RMQ}[\min(\mathsf{ISA}[a], \mathsf{ISA}[b]) + 1\,..\,\max(\mathsf{ISA}[a], \mathsf{ISA}[b])]$ longest common extension

Backward Search of FM-index [6]:

- P : pattern
- R_i: range of the pattern $P[i\,..\,n]$, i.e., $T[\mathsf{SA}[j]\,..\,\mathsf{SA}[j]+i-1] = P[i\,..\,n] \Leftrightarrow j \in R_i$
- $\mathsf{b}(R_i) = \mathsf{C}[P[i]] + \mathsf{BWT}.\mathsf{rank}_{P[i]}(\mathsf{b}(R_{i+1}) - 1) + 1$
- $\mathsf{b}(R_i) = \mathsf{LF}[\mathsf{BWT}.\mathsf{select}_{P[i]}(\mathsf{BWT}.\mathsf{rank}_{P[i]}(\mathsf{b}(R_{i+1}) - 1) + 1)]$
- $\mathsf{e}(R_i) = \mathsf{C}[P[i]] + \mathsf{BWT}.\mathsf{rank}_{P[i]}(\mathsf{e}(R_{i+1}))$

References

[1] D. Benoit, E. D. Demaine, J. I. Munro, R. Raman, V. Raman, and S. S. Rao. Representing trees of higher degree. *Algorithmica*, 43(4):275–292, 2005.

[2] M. Burrows and D. J. Wheeler. A block sorting lossless data compression algorithm. Technical Report 124, Digital Equipment Corporation, Palo Alto, California, 1994.

[3] K. T. Chen, R. H. Fox, and R. C. Lyndon. Free differential calculus, IV. The quotient groups of the lower central series. *Annals of Mathematics*, pages 81–95, 1958.

[4] M. Crochemore and L. Ilie. Computing longest previous factor in linear time and applications. *Inf. Process. Lett.*, 106(2):75–80, 2008.

[5] K. Fan, S. J. Puglisi, W. F. Smyth, and A. Turpin. A new periodicity lemma. *SIAM J. Discrete Math.*, 20(3):656–668, 2006.

[6] P. Ferragina and G. Manzini. Opportunistic data structures with applications. In *Proc. FOCS*, pages 390–398, 2000.

[7] P. Ferragina and G. Manzini. Indexing compressed text. *J. ACM*, 52(4):552–581, 2005.

[8] N. J. Fine and H. S. Wilf. Uniqueness theorems for periodic functions. *Proceedings of the American Mathematical Society*, 16(1):109–114, 1965.

[9] F. Franek, J. Holub, W. F. Smyth, and X. Xiao. Computing quasi suffix arrays. *Journal of Automata, Languages and Combinatorics*, 8(4):593–606, 2003.

[10] M. L. Fredman and D. E. Willard. Surpassing the information theoretic bound with fusion trees. *J. Comput. Syst. Sci.*, 47(3):424–436, 1993.

[11] R. Grossi and J. S. Vitter. Compressed suffix arrays and suffix trees with applications to text indexing and string matching. *SIAM J. Comput.*, 35(2):378–407, 2005.

[12] T. Hagerup. Sorting and searching on the word RAM. In *Proc. STACS*, volume 1373 of *LNCS*, pages 366–398, 1998.

[13] G. Jacobson. Space-efficient static trees and graphs. In *Proc. FOCS*, pages 549–554, 1989.

[14] J. Kärkkäinen, G. Manzini, and S. J. Puglisi. Permuted longest-common-prefix array. In *Proc. CPM*, volume 5577 of *LNCS*, pages 181–192, 2009.

[15] T. Kasai, G. Lee, H. Arimura, S. Arikawa, and K. Park. Linear-time longest-common-prefix computation in suffix arrays and its applications. In *Proc. CPM*, volume 2089 of *LNCS*, pages 181–192, 2001.

[16] S. Lee and K. Park. Dynamic rank/select structures with applications to run-length encoded texts. *Theor. Comput. Sci.*, 410(43):4402–4413, 2009.

[17] U. Manber and E. W. Myers. Suffix arrays: A new method for on-line string searches. *SIAM J. Comput.*, 22(5):935–948, 1993.

[18] J. Sirén. Sampled longest common prefix array. In *Proc. CPM*, volume 6129 of *LNCS*, pages 227–237, 2010.

[19] J. A. Storer and T. G. Szymanski. Data compression via textural substitution. *J. ACM*, 29(4):928–951, 1982.

[20] A. C. Yao. Should tables be sorted? *J. ACM*, 28(3):615–628, 1981.

[21] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Trans. Information Theory*, 23(3):337–343, 1977.

[22] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Trans. Information Theory*, 24(5):530–536, 1978.