FACULTY OF SCIENCE
UNIVERSITY OF COPENHAGEN

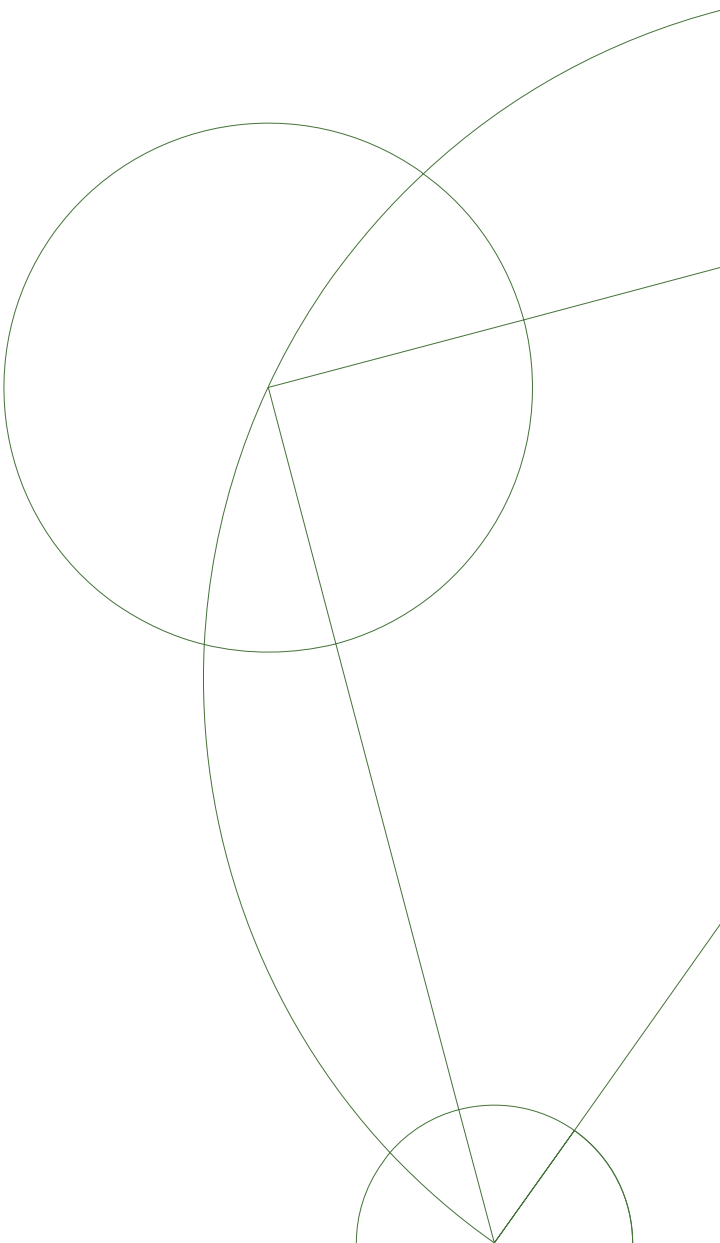
DEPARTMENT OF COMPUTER SCIENCE

Bachelor

Malware detection system using suffix array

31/10-2016

Mark Roland Larsen <fvg932j@alumni.ku.dk>



Supervisors
Michael <gxl387@alumni.ku.dk>
Troels Larsen <gxl387@alumni.ku.dk>

Contents

1 Description

2 Preface

Functional Programmering using F - Exercise A7-1, A7-2 og A7-3

3 Limitations

I følgende opgave arbejdes der på binære træer med typen

4 Introduction

The string matching problem is found in various fields of study. In biology, string matching algorithms significantly aid biologists in retrieving and comparing DNA strings, reconstructing DNA strings from overlapping string fragments and looking for new or presented patterns occurring in a DNA**?**. Text-editing applications also adopt string matching algorithms, whenever the application has to acquire an unambiguous occurrences of a user-given pattern, such as a word in some document**??**. String matching is used in music equipment, AI (artificial intelligence) and in addition, various software applications like virus scanners (anti-virus) or intrusion detection systems, frequently adopt string matching algorithms as a practical tool, to secure data security over the internet **?**. Fundamentally, string matching is a method to find some pattern $P = \{p_1, p_2, \ldots, p_n\}$ in a given text $T = \{t_1, t_2, \ldots, t_m\}$, over some finite alphabet Σ as illustrated in **??** **?**.

5 String Matching

Following the tradition of Given a pattern P and a long text T, the problem consist of finding all occurrences of pattern P, if any, in text T
apalike:gusfield.

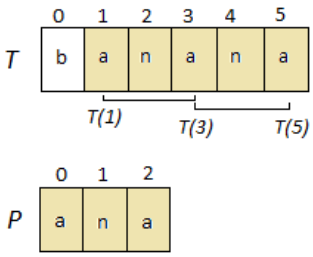


Figure 1: The text T={abattabappab} and pattern P={abap} over the alphabet Σ={abpt}. The pattern P occours only one place, exactly between $T(5, 9)$.

String matching is both an algorithmic problem and data structure problem. The static data structure consist of preprocessing some predefined large text $T = \{t_1, t_2, \ldots, t_m\}$, and query some smaller pattern $P = \{p_1, p_2, ..., p_n\}$ **?**. The objective is to preprocess text

T and query pattern P in text T in linear time, $O(m), m \in |T|$ [1] and $O(n), n \in |P|$, respectively ?.

5.1 Suffix trees

5.2 Suffix arrays

5.3 Operations on suffix arrays

6 Linear time $O(m)$ preprocessing algorithms

6.1 Suffix Array Induced Sorting (SA-IS)

6.2 SA-IS - correctness and compleness

7 Linear time $O(m)$ queuing

7.1 Some Algorithm - Linear time pattern searching

7.2 Some Algorithm - correctness and compleness

8 Malware

8.1 Understanding Malware

8.2 Building database of known malware - SHA1 encryption

8.3 String matching in Malware detection systems

8.4 Building interactive systems - Windows (R) Forms

8.5 Implementing a Malware detection system using preprocessed suffix arrays of known malware

9 Evaluation and recommendations

Jeg har ikke nået at lave denne, men smider alle opgaverne til dig torsdag eller fredag, som jeg skal beskrevet i mailen. Håber det er iorden.

10 Discussion

?

[1] See ?? for a description of algorithmic time analysis

A One

Etiam pede massa, dapibus vitae, rhoncus in, placerat posuere, odio. Vestibulum luctus commodo lacus. Morbi lacus dui, tempor sed, euismod eget, condimentum at, tortor. Phasellus aliquet odio ac lacus tempor faucibus. Praesent sed sem. Praesent iaculis. Cras rhoncus tellus sed justo ullamcorper sagittis. Donec quis orci. Sed ut tortor quis tellus euismod tincidunt. Suspendisse congue nisl eu elit. Aliquam tortor diam, tempus id, tristique eget, sodales vel, nulla. Praesent tellus mi, condimentum sed, viverra at, consectetuer quis, lectus. In auctor vehicula orci. Sed pede sapien, euismod in, suscipit in, pharetra placerat, metus. Vivamus commodo dui non odio. Donec et felis.

References

D. Gusfield, *Algorithms on strings, trees, and sequences : computer science and computational biology*. The Pres Syndicate Of The University Of Cambridge, 1 ed., 1997.

R. L. R. . C. S. Thomas H. Cormen, Charles E. Leiserson, *Introduction To Algorithms*. The MIT Pres, Cambridge, Massachusetts, London, England, 3th ed., 2009.

D.-N. L. Nguyen Le Dang and V. T. Le, "A new multiple-pattern matching algorithm for the network intrusion detection system," *IACSIT International Journal of Engineering and Technology*, vol. 8, no. 2, pp. 1–7, 2016.

"Strings - advanced data structures." https://www.youtube.com/watch?v=F3nbY3hIDLQl.

E. Ju and C. Wagner, "Personal computer adventure games: Their structure, principles, and applicability for training," *ACM SIGMIS Database*, vol. 28, no. 2, pp. 78–92, 1997.

A. Baltra, "Language learning through computer adventure games," *Simulation and Gaming*, vol. 21, pp. 455–452, December 1990.

D. M., "How to use scratch for digital storytelling." https://www.graphite.org/blog/how-to-use-scratch-for-digital-storytelling.

https://www.khanacademy.org/computer-programming/new/pjs.

B. Fry and C. Reas. http://processingjs.org/.

L. K. G. at MIT Media Lab, "Scratch." https://scratch.mit.edu/.

J. E. Ormrod, *Educational Psychology: Developing Learners*. Upper Saddle River, N.J.: Pearson/Merrill Prentice Hall, 5th ed., 2006.

"Programming and problem solving (pop)." http://kurser.ku.dk/course/ndab15009u/2015-2016, 2015/2016.

T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press, third ed., 2009.

S. Denmark, "Cultural habits survey 2012." http://www.dst.dk/en/Statistik/dokumentation/declaration habits-survey.

J. M. Wing, "Computational thinking and thinking about computing." https://www.cs.cmu.edu/afs/cs/usr/wing/www/talks/ct-and-tc-long.pdf, 2008.

E. Alinea, "iskriv." http://iskriv.dk/, 2012.

D. Statistik, "Kvub1204: Children who play computer games by frequency and background." http://www.statistikbanken.dk/KVUB1204, 2015.

T. May and B. K. Walther, *Computerspillets Fortællinger*, vol. 1. Gyldendal, 2013.

L. Blum and T. J. Cortina, "CS4HS: An Outreach Program for High School CS Teachers," *Sigcse '07*, pp. 19–23, 2007.

S. Gray, C. S. Clair, R. James, and J. Mead, "Suggestions for graduated exposure to programming concepts using fading worked examples," *ICER*, pp. 99–110, 2007.

Y. B. Kafai, "Playing and Making Games for Learning: Instructionist and Constructionist Perspectives for Game Studies," *Games and Culture*, vol. 1, no. 1, pp. 36–40, 2006.

T. Nousiainen, *Children's Involvement in the Design of Game-Based Learning Environments*, pp. 49–66. Springer Science, 2009.

J. Moreno-León and G. Robles, "Computer programming as an educational tool in the English classroom," in *2015 IEEE Global Engineering Education Conference*, pp. 961–966, 2015.

J. M. Wing, "Computational thinking and thinking about computing.," *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, vol. 366, no. 1881, pp. 3717–3725, 2008.

J. M. Wing, "Computational thinking," *Communications of the ACM*, vol. 49, no. 3, pp. 33–35, 2006.