FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION

OF HIGHER EDUCATION

ITMO UNIVERSITY

Report

on the practical task No. 8

“Practical analysis of advanced algorithms ”

Performed by

Cheng-Yuan and Ma

Academic group

J4133c

Petr Chunaev *St.*

Petersburg 2022

Language : python

Goal :Practical analysis of advanced algorithms

Problem :

I. Choose two algorithms (interesting to you and not considered in the course)

from the above-mentioned book sections.

II. Analyse the chosen algorithms in terms of time and space complexity, design technique used, etc. Implement the algorithms and produce several experiments. Analyse the results.

Theory:

I chose two algorithm : pattern search algorithm and convex hull algorithm

I chose three different algorithm for both algorithm

I. pattern search algorithm :

Naive Search :

Slide the pattern over text one by one and check for a match. If a match is found, then slide by 1 again to check for subsequent matches.

Time Complexity: O(length of given text^2)

Auxiliary Space: O(1)

the best case will occur when there is no matching pattern in given text .
the worst case will happen when pattern show up in last few words or given text and pattern text are same . It can be O(length of given text * (length of pattern text - length of given text +1))

KMP search :

The basic idea behind KMP's algorithm is: whenever we detect a mismatch (after some matches), we already know some of the characters in the text of the next window. We take advantage of this information to avoid matching the characters that we know will anyway match

this can improve worst case of naive search to O(length of given text)

Boyer Moore Search :

Boyer Moore algorithm starts matching from the last character of the pattern, The main idea behind the efficient approach is that in brute force we have to check every position despite knowing that some positions will never match. So, if we somehow find a way to skip some of the positions that give no useful info we can significantly lower our time used. Before we proceed further we need to know about Bad Character Heuristic and Strong Suffix Heuristic.
In Boyer Moore there are two ways of skipping positions:-1. Bad Character Heuristic 2. Good Suffix Heuristic.

Time Complexity : O(length of given text * length of pattern text)

Auxiliary Space: O(1)

II. convex hull algorithm :

Jarvis algorithm :

The working of Jarvis's march resembles the working of selection sort. In selection sort, in each pass, we find the smallest number and add it to the sorted list. Similarly, in Jarvis's march, we find the *leftmost point* and add it to the convex hull vertices in each pass.

Time Complexity: O(m * n) : n is input node, and m in output node , for every node ,we have to examine all other node to choose next node

Graham algorithm :

This algorithm first sorts the set of points according to their polar angle and scans the points to find the convex hull vertices.

The algorithm takes O(nLogn) time if we use a O(nLogn) sorting algorithm , there are multiple steps in this algorithm , but it use large part of time to sort the nodes .

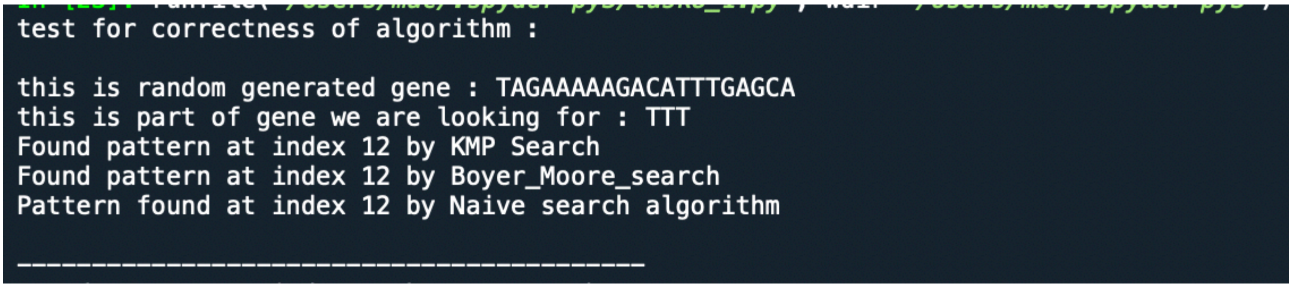
Monotone chain algorithm :

The monotone chain algorithm works in the following steps:

- Sort the points with respect to their x-coordinates (y if there is a tie in the x plane).
- The upper and lower hulls are then calculated in O(n) time.
- Find the leftmost point, rotate clockwise to find the next point and repeat until the right most point is found. Rotate again clockwise to find the lower hull.

The monotone chain algorithm for constructing the convex hull is completed in O(nlogn) time, The upper and lower hulls are calculated in O(n) time.

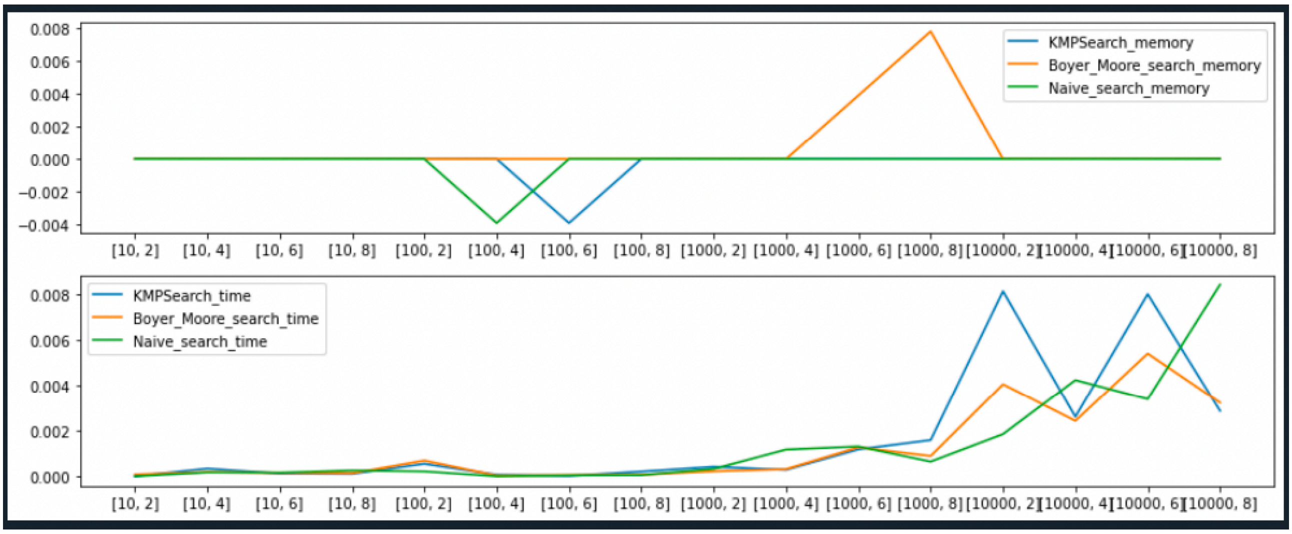
Results:

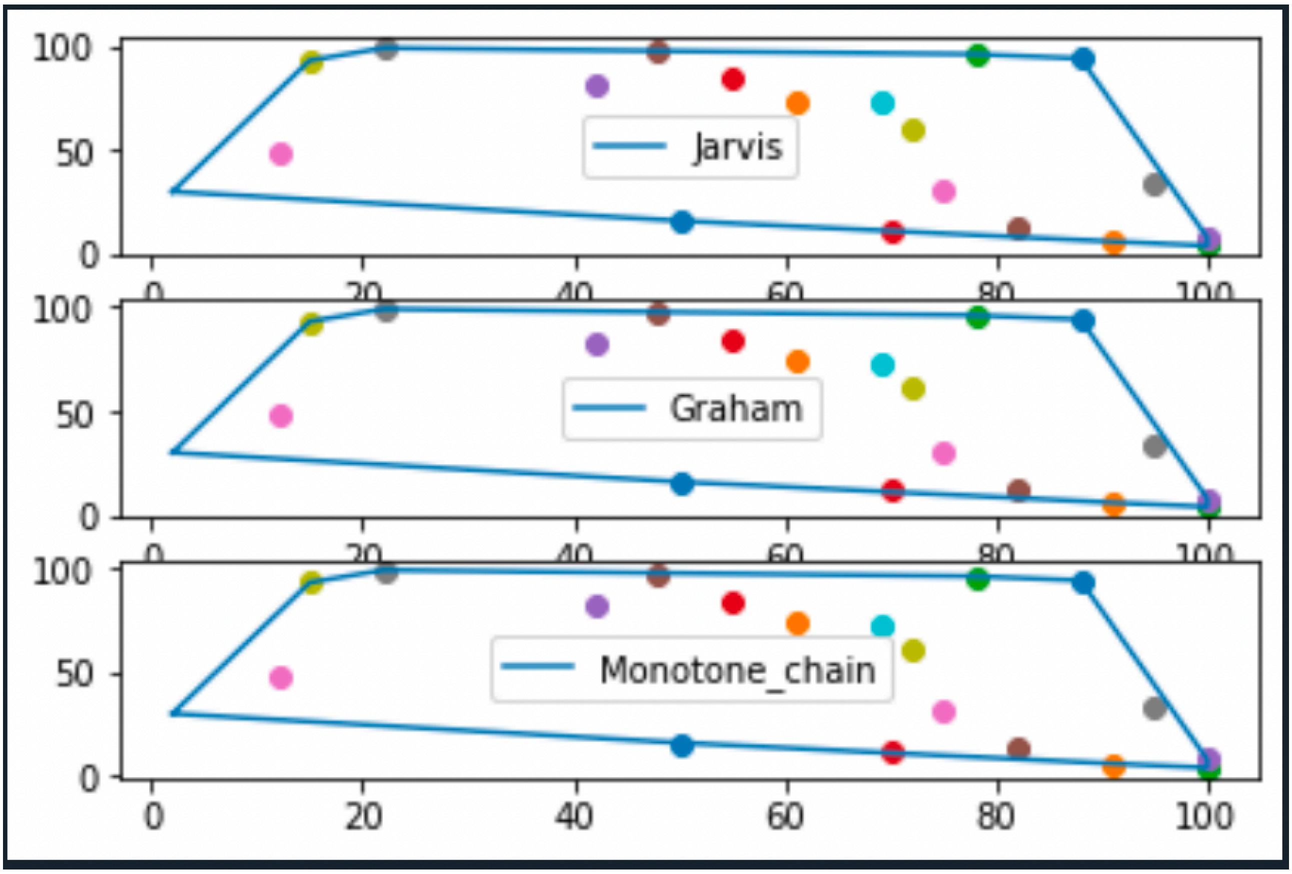


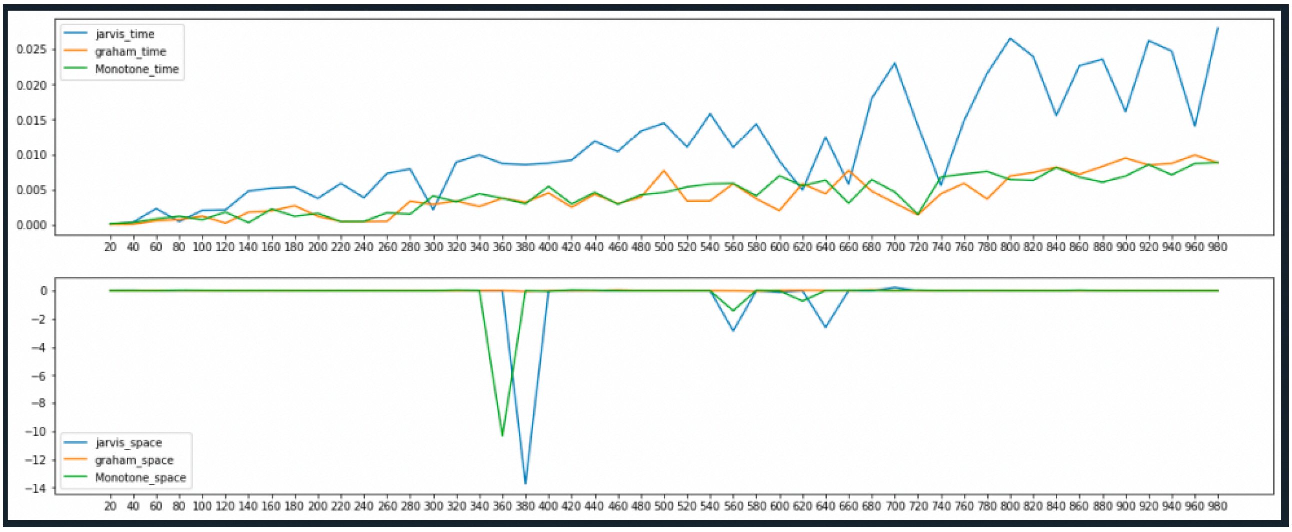
```
test for correctness of algorithm :

this is random generated gene : TAGAAAAAGACATTTGAGCA
this is part of gene we are looking for : TTT
Found pattern at index 12 by KMP Search
Found pattern at index 12 by Boyer_Moore_search
Pattern found at index 12 by Naive search algorithm

-----------------------------------------
```







Conclusion:

Let’s talk about the result from first algorithm , we can see that three of them completed with same result ,which they found pattern at index 12 . As we can see from second graph showing execution time and space , comparing with naive search , other two have inclination to have lower execution time , due to preprocessing process instead of using brute method . Speaking of space complexity , we can see these algorithms do not have to use excess space to execute .

Now we move on to second algorithm , which is convex hull algorithm , from first image we can see algorithms successfully found a convex hull from 20 random nodes and got to same image . Form second image of execution time and space , we also can see space complexity is O(1) . In the other side , execution time has dramatically different result , Jarvis algorithm (gift wrapping) has comparatively larger execution time comparing with other 2 algorithms .

In the real scenario we have problems that need quick and efficient algorithms for computation. The pattern search and matching method are one of these. In the example of web application we deal with various data and its sets, where we have to perform image, audio, text, string, video, etc. types of searching operation . Also convex hull algorithm can be used n the calculation of hyperbolic manifolds and low-dimensional topology. Also in computer vision , their use is widely used for self-driving car .

Appendix:

https://github.com/MaChengYuan/task8.git