[CENG 315 All Sections] Algorithms

Navigation

- Dashboard
  - Site home
  - Site pages
  - My courses
    - 571 - Computer Engineering
      - [CENG 351 All Sections]
      - CENG 300 All Sections
      - CENG 300 Section 4
      - CENG 315 All Sections
        - Participants
        - Badges
        - Competencies
        - Grades
        - General
        - October 13 - October 19
        - October 20 - October 26
        - October 27 - November 2
        - November 3 - November 9
        - November 10 - November 16
        - November 17 - November 23
        - November 24 - November 30
        - December 1 - December 7
        - December 8 - December 14
        - December 15 - December 21
        - December 22 - December 28
        - December 29 - January 4
        - January 5 - January 11
        - January 12 - January 18
        - January 19 - January 25
          - THE 8 Discussion Forum
          - the8 sampleIO
          - **THE 8**
            - Description
            - Submission
            - Edit
            - Submission view
          - Lecture - NP-Completeness
          - Week 14 Annotations
      - CENG 315 Section 3
      - CENG 331 All Sections
      - CENG 331 Section 2
      - CENG 351 Section 3
    - 651 - Music and Fine Arts
    - 612 - Modern Languages (Persian)
    - 642 - Turkish Language

Description | Submission | Edit | Submission view

THE 8

Available from: Friday, January 21, 2022, 11:00 AM
Due date: Friday, January 21, 2022, 11:59 PM
Requested files: the8.cpp (Download)
Type of work: Individual work

Multi-Pattern Matching

In this exam, you are expected to implement a multi-pattern matcher. Provided a text and a vector of patterns, your code should match all occurences of all patterns (in the vector) and return all valid shifts. To be more specific, you are given a string *text* of length N and a vector<string> *patterns* of size K, containing strings of length M. (All patterns have the same length M and are distinct.) You should scan the *text* and match the strings given in the *patterns* vector. You are expected to populate and return a vector<int> that represents the set of valid shifts (taken over all patterns), in increasing order.

An implementation that runs in O(N + K * M + S * M) time (where S is the number of matches) is expected to receive full points in this THE. Such a bound can practically be achieved using a variation of the **Rabin-Karp Algorithm** (with reasonable parameters that keeps the false positive rate down). We highly suggest that you follow this approach.

For this THE, as we expect you to implement the algorithmic details, you are not allowed to use any advanced algorithm libraries. For this reason, **your code is not allowed to include any additional headers/libraries** other than the ones that we already provide you.

Example IO:
****Indices start from 0, as usual.*

1) Given **text = "abcde"** and **patterns ={"ab", "ed"} :**

- Only "ab" has a valid shift (and it has only 1 valid shift). Since it matches the text[0..1] the shift in that case is 0.
- Notice that a valid shift value is the index of the text where the match begins.
- "ed" does not match with any substring of "abcde", so it does not have any valid shift.
- Thus, your return vector must be {0}.

2) Given **text = "abcdefgijbcde"** and **patterns ={"bcde", "defg", "qwer", "uutz"} :**

- "bcde" has two valid shifts: 1 and 9, since it matches text[1..4] and text[9..12].
- Notice that a string may have more than one valid shifts in the given text.
- "defg" has one valid shift: 3, since it matches text[3..6].
- "qwer" and "uutz" have no valid shifts.
- Thus, your return vector must be {1, 3, 9}.

Specifications:

- There is **1 task** to be solved in **12 hours** in this take home exam.
- You will implement your solution in **the8.cpp** file.
- You are free to add other functions to **the8.cpp**
- Do **not** change the first line of **the8.cpp**, which is **#include "the8.h"**
- Do **not** change the arguments and the return value of the function **the8().** You should accumulate valid shift values in the vector<int> shifts and return that vector**.**
- *<iostream>*
  <climits>
  <cmath>
  <string>
  <ctime>
  <string>
  <vector>
  <array>
  <list>
  <forward_list>
  <cmath>
  <random>
  <cstring>
  <cstdlib>
  are **included** in "the8.h" for your convenience.

- **Do not include any other library** and do not write *include* anywhere in your **the8.cpp** file (not even in comments).

- You can evaluate your **the8.cpp** on virtual lab environment. If you click **evaluate**, you will get a feedback for your current work and your work will be **temporarily** graded for **limited** number of inputs.
- Your solutions are evaluated both in terms of correct output and correct running time. To receive credit, your solution should pass both tests.
- The grade you see in lab is **not** your final grade, your code will be reevaluated with **completely different** inputs after the exam.
- **Evaluation may last around 1 minutes** even if your solution is correct, due to huge input sizes.
- If you are sure that your solution works in the expected complexity constrains but your evaluation fails due to limits in the lab environment, the constant factors may be the problem.

Constraints:

- Maximum text size (i.e. max N value) is **20000000** (20 millions).
- All patterns given at one test case are of the same length (i.e. M is fixed.).
- Maximum pattern length (i.e. max M value) is **1000**.
- Maximum number of patterns (i.e. max K value) is **10000**.
- Maximum number of valid shifts (i.e., max S value) is **10000**.
- The text and the patterns to be matched are strings of lowercase English letters.

Requested files

the8.cpp

```cpp
#include "the8.h"

//DO NOT ADD OTHER LIBRARIES

using namespace std;

vector<int> the8(const string& text, const vector<string>& patterns){

    vector<int> shifts; //DO NOT CHANGE THIS

    /*****************
     *
     * YOUR CODE HERE
     *
     * *************/

    return shifts; //DO NOT CHANGE THIS
}

```

VPL