Andy Cao

04/06/2021

dongyicao123@csu.fullerton.edu

Project-1 Report

1. Pseudocode describing your telegraph style string algorithm.

```
String telegraph_style(string s){
        string new_str = " ";                         //create new str for telegraph style
        for(int i = 0; i < s.size(); i++){            //run for loop to search each element
                if(islower(s[i])){                            // lower case situation
                        new_str .push_back(toupper(s));
                }else if(punctuation char !?; ){      // punctuation character situation
                        new_str.push_back('.');       //convert !?; to .
                }else if(isdigit(s[i]) || uppercase char || periods )){ //digit number situation
                        new_str.push_back(s[i]);
                }else if(space){                      //space situation
                        if(new_str != ' '){           //multiple space situation
                                new_str.push_back(' ');
                        }
                }
        if(new_str != "STOP."){               //add if the new_str do not have STOP.
                new_str.add_back("STOP.")
                break;
                }
        }
        return new_str;                       //return new string
}
```

2. Mathematical analyses for each of the three algorithms.

- find_dip algorithms

$T(n) = 1 + n(1+1 +1) + 1$

$T(n) = 2+3n$

Proof: by properties of O

$3n + 2 \in O(3n + 2)$

$= O(max(3n, 2))$

$= O(3n)$

$= O(n)$

- Longest balanced span algorithms

T(n) = 1 + n*n(1+1+1+1+1) +1

T(n) = 2 + 5n^2

Proof: by limits

lim(n→∞) T(n)/ f(n) = lim(n→∞) (5n^2 + 2)/ (n^2)

= lim(n→∞) 5n^2/ n^2 + lim(n→∞) 2/ n

= 5 + 0 = 5

5 which is non-negative and constant with respect to n therefore 5n^2 + 2 ∈ O(n^2)

- Telegraph style string

$T(n) = 1 + n(2 + 1 + 2 + 1 + 1 + 1) + 2 + 1 + 1$

$T(n) = 4+ 8n$

Proof: properties of O

$4-+8n \in O(8n + 4)$

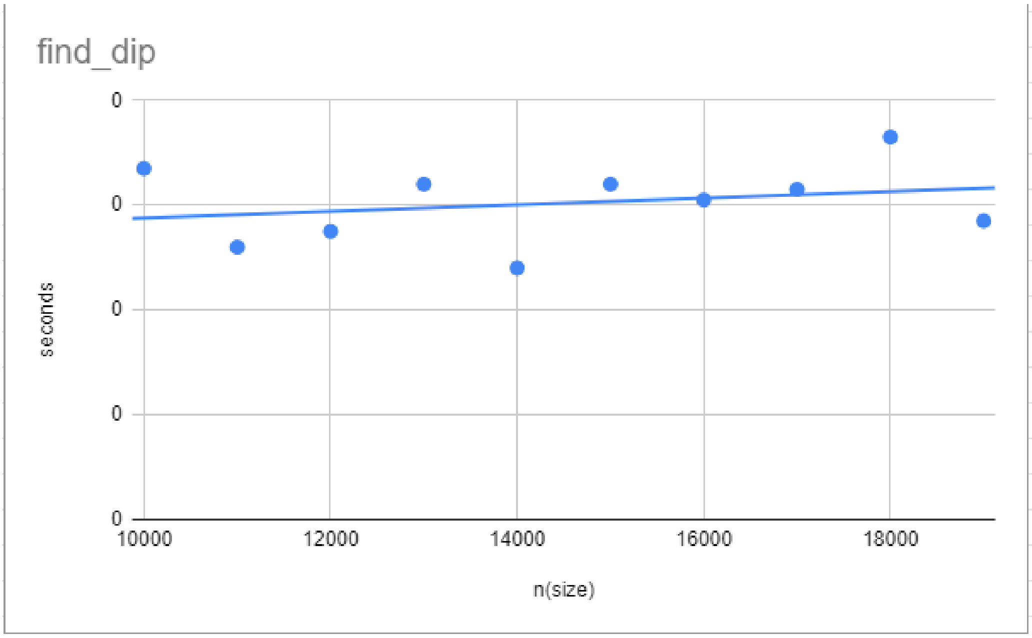
$= O(max(8n, 4))$

$= O(8n)$

$= O(n)$

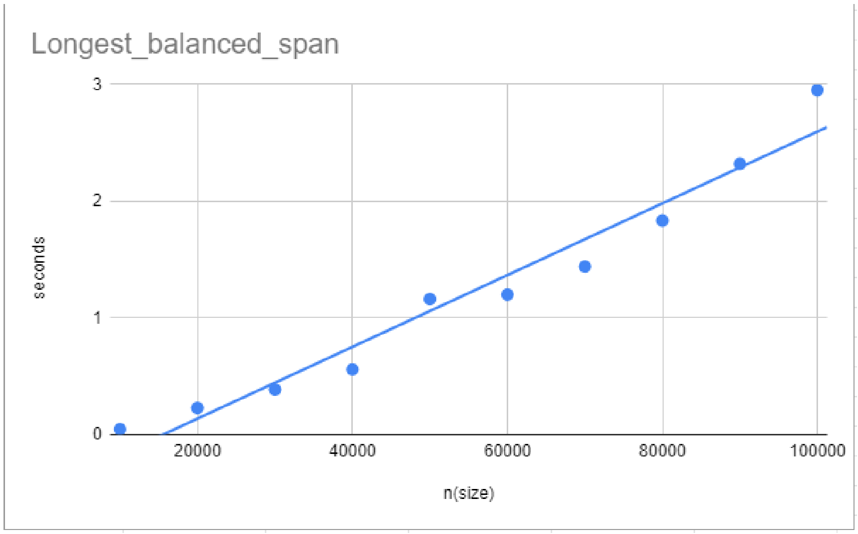
3. Scatter plots for each of the three algorithms.

    - Find_dip



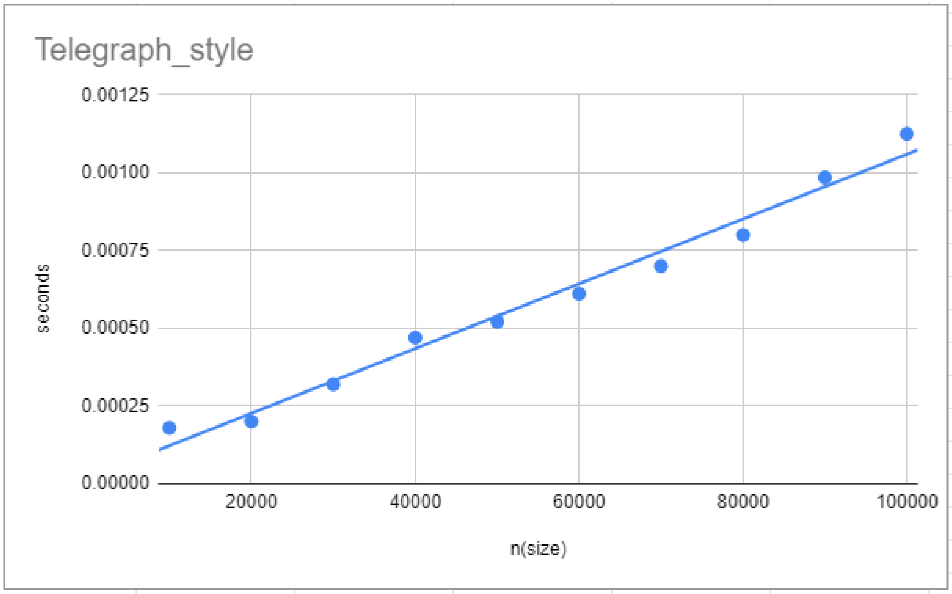
n	seconds
10000	0.000000067
12000	0.000000055
14000	0.000000048
16000	0.000000061
18000	0.000000073
11000	0.000000052
13000	0.000000064
15000	0.000000064
17000	0.000000063
19000	0.000000057

- Longest_balanced_span



n	seconds
10000	0.0463788
20000	0.228574
30000	0.38614
40000	0.558505
50000	1.16179
60000	1.20029
70000	1.43993
80000	1.83446
90000	2.32049
100000	2.95241

- Telegraph Style



n	seconds
10000	0.000179398
20000	0.000200314
30000	0.000319238
40000	0.000469519
50000	0.00052095
60000	0.000610655
70000	0.000699728
80000	0.000799975
90000	0.000984559
100000	0.00112488

4. Answers to the following questions. (Each answer should be at least one complete sentence.)

    1. What is the efficiency class of each of the algorithms, according to your own mathematical analysis? (You are not required to include all your math work, just state the classes you derived and proved.)

- Time complexity of find_dip: O(n)
- Time complexity of longest_balanced_span: O(n^2)
- Time complexity of telegraph_style: O(n)

2. Between the dip search and longest balanced span algorithms, is there a noticeable difference in the running speed? Which is faster, and by how much? Does this surprise you?

Between the dip search and longest balanced span algorithms, the dip search would run faster because dip search takes O(n) and longest balanced span algorithms takes O(n^2) run time.because O(n^2) use takes nested for loop so the size gets larger and the time gets slower.

3. Are the fit lines on your scatter plots consistent with the efficiency classes predicted by your math analyses? Justify your answer.

For the longest balanced span and telegraph style algorithms are consistent with the efficiency classes predicted by the math analyses, because the longest balanced span takes O(n^2) which gets slower when the size gets larger. Same as the telegraph style algorithms. But for the dip search algorithms, the scatter plot seems different to the time complexity. When the size gets larger, the time runs almost the same seconds.

4. Is all this evidence consistent or inconsistent with the hypothesis stated on the first page? Justify your answer.

All the evidence is consistent with the hypothesis stated on the first page, because these three algorithms' time complexity are almost the same as the scatter plots's run time. The longest balanced span would take more time than the other two algorithms because of the time complexity is O(n^2) which is proof by the scatter plots and mathematical analysis.