BLG336E, Analysis of Algorithms II, Spring 2017 Project Report 2

Name: Emre Özdil

Student ID: 150120138

[15 points] If the advantages of divide and conquer methodology were not to be used in your implementation, how would you formulate the problem?

If we choose brute force, complexity is O(n^3). Therefore, we have a time problem. Algorithm run time is so slow.

[15 points] What parameters does the complexity of your implementation depend on? How would you represent the worst case complexity in big O notation?

n = the number of word

For divide complexity->n

For conquer complexity->n

Thus, total complexity is O(n^2)

[20 points] If the rule VP → NP VP were to be added to current set of grammar rules, this would cause ambiguity because it would overlap with the rule S → NP VP. Briefly explain, how you would change your implementation to overcome this problem. Would it change the worst case complexity? If so, what would the complexity be in big O notation?

If we have two tag, program use S->NP VP.

Other case, VP->NP VP.

Example: S[NP(x)VP[NP[(y)VP(z)]]

Finally, we add one condition; thus, complexity is not change.

Complexity is O(n^2)