SAT ANALYSIS TASK 2

Anghelo De La Cruz

August 21, 2017

Contents

1 NEXT Question 1

- Using mathematical notation and computer science conventions describe the time complexity of the algorithm MinOP for the worst case.

- What are the naive elements of the algorithm MinOP that allow a specific design pattern to be applied to improve its time complexity?

- Describe in detail the aspects of the methodology you would use for the design of the improved algortithm.

2 TODO Question 2

- Write in pseudocode your own improved time complexity version of this algorithm. This algorithm should use meaningful variable names, it should indicate what inputs and outputs are expected, as well as including comments for complicated commands. Reference to external procedures or functions that are called should include a measure of their time complexity.

3 TODO Question 3

- Describe the relationship in a methodical and systematic way using mathematrcal notation and compuer science conventions for the time complexity of the new version of algorithm with regards to its input size and its asymptotic running time in the best and worst cases.

- Desribe the situations where the best case time complexity will occur and contrast it with the type of input that results in the worst case time complexity for the new algorithm that you created.

4 TODO Question 4

- Compare your new algorithm with the orginal MinOP algorithm. Discuss and highlight the improvements that your new version has made and specify any trade-offs that you made to achieve the improvement in time complexity.

- Create a graph of the best and worst case time complexity against the input size for the orginal algorithm and your improved version. Clearly label the axes and the lines representing each algorithm on your graph by either labelling the line or using a legend.

- Justify the correctness of your new improved algorithm, using a valid argument. Diagrams may also be used to illustrate your argument.

5 Solution 1

5.1 Part (a)

5.2 Part (b)

The algorithm MinOP contains reoccuring sub-problems that allow recursion to be used to solve this problem.

5.3 Part (c)

There are two ways I would improve tihs algorithm:

- Backtracking
- Dyanamic Programming

6 Solution 2

[n, Operator, Operand, MinCount] afds