Computational Complexity

K. Subramani, LCSEE, West Virginia University

February 2, 2015

1 Review of concepts

1. Alphabet, strings and languages.
2. Problems.
3. Connection between problems and languages.
4. Asymptotics.
5. Calculus for computation.
6. Probability and random variables.
7. Abstract algebra.
8. Upper and lower bounds (maximum and minimum).
9. Problem paradigms (Undecidability).

2 Problems and Solutions

1. Definition of problem. (Chess paradox).
2. Decision and Search problems.
3. Solutions and Algorithms.
4. A discussion of the GCD problem.
5. Euclid's observation.
6. Recursive algorithm.
7. Analysis of algorithms - Worst-case, average case, notion of adversary.

3 Time, Space and Scaling

1. Notion of time.
2. Notion of space.
3. Main question: How do these resources scale with increasing input size?
4. Analyzing Euclid's algorithm.

4 Intrinsic Complexity

1. Definition.
2. Analyzing the two integer multiplication problem.
3. Bringing the running time down from $O(n^2)$ to $O(n^{2.56})$.

5 Polynomial time

1. Polynomial versus exponential.
2. Details of analysis focusing on Euler's algorithm.
3. The class **P**.
4. Defining **TIME**$(f(n))$.
5. The function $n^{\log n}$.
6. The class **EXP**.
7. Motivation for the class **P**.
8. Robustness of **P**.
9. Binary/unary encoding. Buffing up the input.
10. **P**'s Imperviousness to details.
11. Tractability and mathematical insight.