Section A

- Everything included in IOI syllabus
- Elementary Number Theory
- Graph Theory and Algorithms
- Enumerative Combinatorics
- Probability
- Geometry

The main focus will be on the following aspects:

1. Systematically following, simulating and reasoning about sets of instructions, protocols, structures, etc.
2. Understanding the correctness of algorithms
3. Assessing performance of algorithms
4. Reasoning about discrete structures
5. Reasoning about combinatorial games
6. Understanding implications of logical statements

Section B

- Algorithms:
    - Graph algorithms (connectivity, spanning trees, matchings, flows etc.)
    - Number-theoretic algorithms (primality testing, factorization etc.)
    - Computational geometry
    - Divide and conquer, dynamic programming, greedy algorithms, and other common techniques
    - Basic running time analysis
    - Randomized and approximation algorithms
- Complexity:

    - Basic complexity classes (P, NP, P-space etc.)
    - Reductions and completeness
    - Interactive proofs, probabilistically checkable proofs
    - Hardness of approximation
- Theory of Computation:
    - DFA/NFA and regular languages
    - Context-free grammars and pushdown automata
    - Turing machines / Oracle Turing machines
- Discrete Mathematics:
    - Graph theory
    - Enumerative combinatorics
    - Probability
- Logic:
    - Propositional logic
    - First-order logic
    - Truth tables
    - Proof systems
- Miscellaneous:
    - Game theory
    - Basic programming in a language of choice
    - Computational number theory
    - Derandomization techniques
    - Cryptography
    - Quantum information and computation
    - Linear algebra

The main focus will be on the following aspects:

1. Comprehensive understanding of algorithms and algorithmic paradigms such as greedy algorithms, dynamic programming, divide & conquer, and introductory graph algorithms. A preliminary knowledge of analysis of these algorithms is essential.
2. Understanding of data structures and various discrete structures such as graphs, trees, heaps, stacks, and queues.
3. An understanding of finite state machines, pushdown automata, and Turing machines, along with their properties and representations including grammars and computation models.
4. An understanding of computation in terms of complexity and decidability.