

ขอบเขตเนื้อหาที่ใช้ในการแข่งขัน
โครงการแข่งขันแก้ไขปัญหาด้วยการเขียนโปรแกรมคอมพิวเตอร์
ระดับมัธยมศึกษาตอนปลาย ปีการศึกษา 2567 ครั้งที่ 2

แบ่งได้เป็น 3 หมวด คือ (1) คณิตศาสตร์ (2) พื้นฐานวิทยาการคอมพิวเตอร์ และ (3) อัลกอริทึม

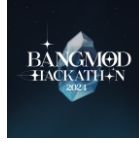
1. หมวดคณิตศาสตร์

1.1. เลขคณิตและเรขาคณิต

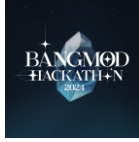
- 1.1.1. จำนวนเต็ม คุณสมบัติของเลขจำนวนเต็ม (ค่าบวก ค่าลบ เลขคู่ เลขคี่ การหารลงตัว จำนวนเฉพาะ)
- 1.1.2. เลขเศษส่วน และร้อยละ
- 1.1.3. จุด เวกเตอร์ พิกัดจุดแบบคาร์ทีเซียน (Cartesian coordinates) ในตารางสองมิติที่มีพิกัดเป็นจำนวนเต็ม
- 1.1.4. ระยะทางแบบยูคลิด ทฤษฎีพีทาโกรัส
- 1.1.5. ส่วนของเส้นตรง จุดตัดของเส้นตรง และคุณสมบัติพื้นฐานที่เกี่ยวข้อง
- 1.1.6. มุม สามเหลี่ยม สี่เหลี่ยมผืนผ้า สี่เหลี่ยมจัตุรัส วงกลม

1.2. โครงสร้างไม่ต่อเนื่อง (discrete structures)

- 1.2.1. ฟังก์ชัน ความสัมพันธ์ และเซต
- 1.2.2. ตรรกศาสตร์พื้นฐาน
- 1.2.3. วิธีการพิสูจน์
- 1.2.4. วิธีการนับเบื้องต้น
 - 1.2.4.1. กฎของการบวกและกฎของการคูณ (Sum rule and Product rule), หลักการเพิ่มเข้า-ตัดออก (inclusion-exclusion principle), ลำดับเลขคณิตและเรขาคณิต จำนวนแบบฟีโบนัชชี (Fibonacci numbers)
 - 1.2.4.2. กฎรังนกพิราบ (Pigeonhole principle) เพื่อใช้ในการหาขอบเขต
 - 1.2.4.3. การเรียงสับเปลี่ยน และวิธีจัดหมู่ระดับพื้นฐาน
 - 1.2.4.4. ฟังก์ชันเลขเศษส่วน (Fractional function) และสัมประสิทธิ์ทวินาม (Binomial coefficient)
- 1.2.5. กราฟและต้นไม้
 - 1.2.5.1. ต้นไม้และคุณสมบัติพื้นฐาน
 - 1.2.5.2. กราฟไม่มีทิศทาง (degree, path, cycle, connectedness, Handshaking Lemma)
 - 1.2.5.3. กราฟแบบมีทิศทาง (in-degree, out-degree, directed path/cycle)
 - 1.2.5.4. Spanning trees
 - 1.2.5.5. วิธีการเดินผ่านต้นไม้ (traversal strategies: defining the node order for ordered trees)
 - 1.2.5.6. 'Decorated' graphs with edge/node labels, weights, colors
 - 1.2.5.7. Multigraphs และ graphs ที่มี self loops
 - 1.2.5.8. planar graphs, bipartite graphs, และ hypergraphs



- 1.3. เนื้อหาเพิ่มเติม
 - 1.3.1. แคลคูลัส
 - 1.3.2. ความน่าจะเป็น
 - 1.3.3. สถิติ
 - 1.3.4. จำนวนจริงและจำนวนเชิงซ้อน
 - 1.3.5. ภาควัดกรวยทั่วไป (parabolas, hyperbolas, ellipses)
 - 1.3.6. โพลิกอน
2. หมวดพื้นฐานวิทยาการคอมพิวเตอร์
 - 2.1. พื้นฐานด้านการเขียนโปรแกรม
 - 2.2. ทักษะการแก้ปัญหา (problem-solving skill)
 - 2.3. พื้นฐานโครงสร้างข้อมูล
 - 2.3.1. ชนิดข้อมูลดั้งเดิม (Primitive data type) ได้แก่ Boolean, signed/unsigned integer, character
 - 2.3.2. แถวลำดับ (อาร์เรย์ อาร์เรย์หลายมิติ)
 - 2.3.3. Record/Struct
 - 2.3.4. สตริงและการดำเนินการกับสตริง
 - 2.3.5. Static และ Stack allocation
 - 2.3.6. Lined structures (ทั้งที่เป็นแบบเส้นตรง และแบบที่แบ่งเป็นสาขาได้)
 - 2.3.7. การสร้าง โครงสร้างกองซ้อน (stack), คิว (queue), ต้นไม้ และกราฟ
 - 2.3.8. การเลือกโครงสร้างข้อมูลที่เหมาะสม
 - 2.3.9. คิวลำดับความสำคัญ (priority queue), ไดนามิกเซต (dynamic set), ไดนามิกแมพ (dynamic map)
 - 2.4. การเรียกตัวเองซ้ำ (Recursion)
 - 2.4.1. แนวคิด
 - 2.4.2. ฟังก์ชันทางคณิตศาสตร์ที่เรียกตัวเองซ้ำ
 - 2.4.3. วิธีแบ่งแยกและเอาชนะ (divide and conquer)
 - 2.4.4. อัลกอริทึมการย้อนรอยแบบเรียกตัวเองซ้ำ (recursive backtracking)
3. หมวดอัลกอริทึม
 - 3.1. พื้นฐานการวิเคราะห์ความซับซ้อนของอัลกอริทึม (algorithmic complexity)
 - 3.2. กลวิธีทางอัลกอริทึม
 - 3.2.1. Brute-Force algorithm
 - 3.2.2. Greedy algorithm
 - 3.2.3. การแบ่งแยกและเอาชนะ
 - 3.2.4. Backtracking (ทั้งที่เป็นแบบเรียกตัวเองซ้ำ และไม่เรียกตัวเองซ้ำ)
 - 3.2.5. Branch-and-Bound algorithm
 - 3.2.6. Pattern matching and string/text algorithm
 - 3.2.7. Dynamic programming



3.3. อัลกอริทึมเชิงคำนวณพื้นฐาน

- 3.3.1. อัลกอริทึมเชิงตัวเลขพื้นฐานที่เกี่ยวข้องกับจำนวนเต็ม เช่น Radix Conversion, Euclid's algorithm, Primality test in $O(\sqrt{N})$, Sieve of Eratosthenes, Factorization, Efficient exponentiation
- 3.3.2. การจัดการอาร์เรย์ขั้นพื้นฐาน (รวมถึงการทำฮีโธแกรม และ Bucket sort)
- 3.3.3. Sequential และ Binary search
- 3.3.4. Search by elimination
- 3.3.5. การแบ่งข้อมูล (partitioning) การจัดลำดับด้วยการแบ่งข้อมูลซ้ำๆ Quick sort
- 3.3.6. การเรียงข้อมูลที่มีเวลาที่แย่ที่สุดเป็น $O(N\log N)$ เช่น Heap sort และ Merge sort
- 3.3.7. Binary heap พื้นฐาน และ Binary search tree
- 3.3.8. การบรรยายโครงสร้างกราฟ เช่น adjacency list และ adjacency matrix
- 3.3.9. Depth-first and breadth-first traversals of graphs และการหาองค์ประกอบที่เชื่อมต่อกันของกราฟแบบไม่มีทิศทาง
- 3.3.10. Shortest path algorithm เช่น Dijkstra, Bellman-Ford และ Floyd-Warshall
- 3.3.11. Transitive closure (Floyd's algorithm)
- 3.3.12. Minimum spanning tree
- 3.3.13. Topological sort