



หลักสูตรวิชาคอมพิวเตอร์โอลิมปิก สอวน.

สำหรับการสอบคัดเลือกและอบรมค่าย สอวน. วิชาคอมพิวเตอร์

ปรับปรุงล่าสุด ตามที่ประชุมกลุ่มย่อยวิชาคอมพิวเตอร์ วันที่ 26 มิถุนายน 2569

เอกสารฉบับนี้สรุปขอบเขตเนื้อหาสำหรับ TOI-Zero, ค่าย 1, ค่าย 2, TOI และการต่อยอดสู่ IOI

สารบัญ

ภาพรวมหลักสูตร	2
TOI-Zero และเกณฑ์ความพร้อมก่อนค่าย 1	3
ประเภทโจทย์ใน TOI-Zero	3
เกณฑ์ผ่านและความพร้อมก่อนสมัครค่าย 1	3
เนื้อหาการอบรมวิชาคอมพิวเตอร์	3
เรื่องที่จะไม่สอน	9

ภาพรวมหลักสูตร

หลักสูตรสาขาคอมพิวเตอร์ ปี 2569 กำหนดขอบเขตเนื้อหาสำหรับการเตรียมความพร้อมนักเรียนตั้งแต่ระดับเริ่มต้น การคัดเลือกเข้าค่าย 1 การอบรมค่าย 1 การอบรมค่าย 2 ไปจนถึงการแข่งขันคอมพิวเตอร์โอลิมปิกระดับชาติและการต่อยอดสู่การแข่งขันคอมพิวเตอร์โอลิมปิกระหว่างประเทศ โดยมุ่งพัฒนาทักษะการคิดเชิงอัลกอริทึม การแก้ปัญหาด้วยโปรแกรม และการใช้เครื่องมือการแข่งขันอย่างเหมาะสม

ชื่อย่อและคำสำคัญที่ใช้ในเอกสารนี้:

- สอวน. = มูลนิธิส่งเสริมโอลิมปิกวิชาการและพัฒนามาตรฐานวิทยาศาสตร์ศึกษา ในพระอุปถัมภ์สมเด็จพระเจ้าพี่นางเธอ เจ้าฟ้ากัลยาณิวัฒนา กรมหลวงนราธิวาสราชนครินทร์
- TOI = Thailand Olympiad in Informatics หรือการแข่งขันคอมพิวเตอร์โอลิมปิกระดับชาติ
- IOI = International Olympiad in Informatics หรือการแข่งขันคอมพิวเตอร์โอลิมปิกระหว่างประเทศ
- TOI-Zero = ระบบและชุดโจทย์เตรียมความพร้อมก่อนสมัครเข้าค่าย 1 สอวน. วิชาคอมพิวเตอร์
- Grader = ระบบตรวจคำตอบอัตโนมัติสำหรับโจทย์เขียนโปรแกรม
- C/C++ และ Python = ภาษาโปรแกรมที่ใช้ในบริบทของหลักสูตรนี้

หลักสูตรแบ่งออกเป็น 4 ช่วง คือ

1. TOI-Zero
2. การอบรมค่าย 1 (ค่าย 1)
3. การอบรมค่าย 2 (ค่าย 2)
4. การแข่งขันคอมพิวเตอร์โอลิมปิกระดับชาติ และการต่อยอดสู่การแข่งขันคอมพิวเตอร์โอลิมปิกระหว่างประเทศ

เนื้อหาทั้งหมดในทุกช่วงแสดงรวมกันในตารางเนื้อหาการอบรมวิชาคอมพิวเตอร์ โดยมีแนวคิดของแต่ละช่วง ดังนี้

TOI-Zero จะให้ความสำคัญกับพื้นฐานการเขียนโปรแกรมคอมพิวเตอร์ ได้แก่ การใช้ตัวแปร การกำหนดค่าตัวแปร การใช้พจน์คณิตศาสตร์และตรรกศาสตร์ในโปรแกรม คำสั่งแบบมีเงื่อนไข การวนซ้ำ การใช้อาร์เรย์หรือรายการข้อมูล พื้นฐาน พื้นฐานคณิตศาสตร์ที่เกี่ยวข้อง การฝึกแก้โจทย์ด้วยตนเอง และการใช้งานระบบตรวจคำตอบอัตโนมัติของ TOI

ค่าย 1 มุ่งให้ผู้เรียนมีทักษะการเขียนโปรแกรมและการแก้โจทย์แข่งขันด้วยภาษา C/C++ สามารถใช้โครงสร้างข้อมูลพื้นฐาน เช่น array, string, stack, queue และ list ได้เหมาะสม เข้าใจคณิตศาสตร์พื้นฐานที่จำเป็น และเริ่มวิเคราะห์โจทย์อย่างเป็นระบบ

ค่าย 2 มุ่งให้ผู้เรียนเข้าใจคณิตศาสตร์ไม่ต่อเนื่อง โครงสร้างข้อมูล และขั้นตอนวิธีที่จำเป็นต่อการแข่งขันระดับสูง โดยแยกเนื้อหาเป็นทั้งส่วนบังคับและส่วนทางเลือกตามศักยภาพของศูนย์และพื้นฐานของผู้เรียน

TOI & IOI มุ่งให้ผู้เรียนสามารถประยุกต์ใช้อัลกอริทึม โครงสร้างข้อมูล และเทคนิคการพิสูจน์หรือวิเคราะห์ความถูกต้องของวิธีแก้ปัญหาในโจทย์แข่งขันที่ซับซ้อนขึ้น รวมถึงเตรียมพื้นฐานสำหรับการเรียนรู้หัวข้อขั้นสูงที่อาจใช้ในการคัดเลือกหรือฝึกซ้อมเพื่อการแข่งขันคอมพิวเตอร์โอลิมปิกระหว่างประเทศ

การดำเนินการอบรมในแต่ละช่วงควรมุ่งเน้นการแก้ปัญหาด้วยอัลกอริทึมผ่านการเขียนโปรแกรมจริง เนื้อหาในตารางหลักจึงใช้เป็นกรอบอ้างอิงร่วม โดยศูนย์อบรมสามารถปรับลำดับ วิธีสอน ความลึก และโจทย์ฝึกให้เหมาะสมกับสมรรถนะของนักเรียน ระยะเวลาอบรม และข้อจำกัดของแต่ละศูนย์ เพื่อให้ให้นักเรียนมีความพร้อมสูงสุดสำหรับการแข่งขันและการเรียนรู้ต่อเนื่อง

TOI-Zero และเกณฑ์ความพร้อมก่อนค่าย 1

TOI-Zero เป็นระดับเริ่มต้นอย่างเป็นทางการสำหรับนักเรียนที่สนใจสมัครเข้าค่าย 1 สอน. วิชาคอมพิวเตอร์ โดยมุ่งให้ผู้เรียนเริ่มฝึกเขียนโปรแกรมด้วยตนเอง ใช้งานระบบตรวจคำตอบอัตโนมัติได้ เข้าใจรูปแบบโจทย์แข่งขันเบื้องต้น และมีพื้นฐานพร้อมสำหรับการเรียนต่อด้วยภาษา C/C++ ในค่ายถัดไป

ประเภทโจทย์ใน TOI-Zero

ตารางประเภทโจทย์ใน TOI-Zero แสดงขอบเขตลักษณะโจทย์ ทักษะหลัก และตัวอย่างรูปแบบโจทย์ที่ใช้สำหรับเตรียมความพร้อมก่อนสมัครเข้าค่าย 1

ตารางที่ 1: ประเภทโจทย์ใน TOI-Zero

ระดับ	ลักษณะโจทย์	ทักษะหลัก	ตัวอย่างโจทย์
A1	โจทย์ที่รับข้อมูลขนาดเล็กและให้ผลลัพธ์จากสูตรหรือเงื่อนไขโดยตรง	รับค่า แสดงผล ตัวแปร นิพจน์ เงื่อนไขพื้นฐาน และการวนซ้ำจำนวนรอบไม่ซับซ้อน	คำนวณตามสูตร ตรวจสอบ/ไม่ผ่าน นับจำนวนรอบ พิมพ์ผลตามรูปแบบ
A2	โจทย์ที่ต้องประมวลผลชุดข้อมูลหรือหลายกรณีตามเงื่อนไขที่กำหนด	เงื่อนไขหลายกรณี การวนซ้ำอาร์เรย์หรือรายการข้อมูลหนึ่งมิติ การสะสมค่า การกรองข้อมูล ฟังก์ชัน และการจัดรูปแบบผลลัพธ์	ค่าเฉลี่ยตามเงื่อนไข หาค่าสูงสุด/ต่ำสุด นับข้อมูลที่ตรงเงื่อนไข แปลงหน่วยหรือจัดกลุ่มข้อมูล
A3	โจทย์ที่ต้องผสมหลายขั้นตอน เช่น เงื่อนไขซ้อน การวนซ้ำซ้อน หรือการประมวลผลข้อมูลหลายมิติในขอบเขตพื้นฐาน	เงื่อนไขซ้อน การวนซ้ำซ้อน อาร์เรย์หรือรายการข้อมูล การคำนวณหลายขั้น และการระวางกรณีขอบเขต	จำลองขั้นตอนตามกติกา คำนวณคะแนนตามหลายเงื่อนไข ประมวลผลตารางขนาดเล็ก หรือเลือกคำตอบตามเงื่อนไขหลายขั้น

โจทย์ TOI-Zero ควรอยู่ในขอบเขตการเขียนโปรแกรมเบื้องต้น ไม่เน้น algorithm และ data structure ที่ซับซ้อน จุดสำคัญคือให้นักเรียนอ่านโจทย์เอง เขียนโปรแกรมเอง ส่งคำตอบเอง และแก้ไขจากผลตรวจเองได้

เกณฑ์ผ่านและความพร้อมก่อนสมัครค่าย 1

- ทำโจทย์ระดับ A2 และ A3 รวมกันอย่างน้อย 20 ข้อ
- ทำโจทย์ทุกระดับรวมกัน A1, A2 และ A3 อย่างน้อย 40 ข้อ
- แต่ละข้อที่นับผลควรได้คะแนนไม่น้อยกว่า 80 คะแนนจาก 100 คะแนนเต็ม
- ผู้เรียนควรมีความพร้อมในการอ่านโจทย์ เขียนโปรแกรม ส่งคำตอบ ตรวจสอบผล และแก้ไขข้อผิดพลาดด้วยตนเองก่อนเข้าสู่การเรียนในค่าย 1

เนื้อหาการอบรมวิชาคอมพิวเตอร์

ตารางอ้างอิง IOI แสดงความหมายของเลขอ้างอิงและหมวด IOI ที่ใช้ในตารางเนื้อหาการอบรม ส่วนตารางเนื้อหาการอบรมแสดงหัวข้อเนื้อหาและระดับหลักสูตรที่ควรบรรจุในแต่ละช่วง รายการเฉพาะของไทยหรือ TOI-Zero ใช้ TOI-Zero เป็นอ้างอิง

สัญลักษณ์ในคอลัมน์ระดับหลักสูตร: ✓ = เนื้อหาบังคับ/ควรสอน, △ = เนื้อหาทางเลือกหรือเนื้อหาเสริมตามศักยภาพ
ศูนย์และผู้เรียน, ช่องว่าง = ไม่อยู่ในระดับนั้น

ตารางที่ 2: คำอธิบายอ้างอิง IOI ในตารางหลัก

อ้างอิง IOI	หมวด IOI
5.1	Arithmetic and Geometry
5.2	Discrete Structures
6.1	Programming Fundamentals
6.1 PF3 / 6.2 AL3b	Fundamental Data Structures
6.1 PF2 / 6.2	Problem Solving, Algorithms, and Complexity
6.2 AL10	Geometric Algorithms
7 SE3 / 8	Development Tools and Environments
TOI-Zero / 7 SE6	Grader and Validation
7	Software Engineering Practices

ตารางที่ 3: เนื้อหาการอบรมวิชาคอมพิวเตอร์

เนื้อหา	TOI-Zero	ค่าย 1	ค่าย 2	TOI/IOI
5.1 — Arithmetic and Geometry	✓	✓	✓	✓
จำนวนเต็ม การดำเนินการ การยกกำลัง และการเปรียบเทียบ				
สมบัติพื้นฐานของจำนวนเต็ม เช่น เครื่องหมาย ความเป็นคู่/คี่ และการหารลงตัว	✓	✓	✓	✓
เลขคณิตมอดุลาร์พื้นฐาน: การบวก การลบ และการคูณ	✓	✓	✓	✓
จำนวนเฉพาะ	✓	✓	✓	✓
เศษส่วนและร้อยละ	✓	✓	✓	✓
เส้นตรง ส่วนของเส้นตรง มุม สามเหลี่ยม สี่เหลี่ยมผืนผ้า สี่เหลี่ยมจัตุรัส และวงกลม	✓	✓	✓	✓
ระยะทางแบบยุคลิด	✓	✓	✓	✓
ทฤษฎีบทพีทาโกรัส	✓	✓	✓	✓
จุด เวกเตอร์ และพิกัดบนระนาบ		✓	✓	✓
รูปหลายเหลี่ยม: จุดยอด ด้าน ความเป็นรูปอย่างง่าย ความนูน ภายใน และพื้นที่		✓	✓	✓
ลำดับ	✓	✓	✓	✓
อนุกรม		✓	✓	✓
สมการและอสมการ	✓	✓	✓	✓
เมทริกซ์เบื้องต้น		✓	△	✓

ตารางที่ 3: เนื้อหาการอบรมวิชาคอมพิวเตอร์ (ต่อ)

เนื้อหา	TOI-Zero	ค่าย 1	ค่าย 2	TOI/IOI
5.2 — Discrete Structures	✓	✓	✓	✓
ตรรกศาสตร์พื้นฐาน ตัวเชื่อมทางตรรกะ และตารางค่าความจริง				
ตรรกศาสตร์อันดับหนึ่ง ตัวบ่งปริมาณ forall/exists และ modus ponens/modus tollens		✓	△	✓
เซต การเป็นสับเซต คอมพลีเมนต์ ผลคูณคาร์ทีเซียน และ power set	✓	✓	✓	✓
ฟังก์ชันและความสัมพันธ์ รวมถึง inverse, composition, equivalence และ order relations	✓	✓	✓	✓
implication, converse, inverse, contrapositive, negation และ contradiction		✓	△	✓
การพิสูจน์โดยตรง counterexample contraposition contradiction และอุปนัยเชิงคณิตศาสตร์		✓	△	✓
strong induction และนิยามเชิงเรียกซ้ำทางคณิตศาสตร์			△	✓
หลักการนับ: rule of sum/product, ลำดับเลขคณิต/เรขาคณิต และ Fibonacci	✓	✓	✓	✓
permutations, combinations, factorial และ binomial coefficients	✓	✓	✓	✓
หลักบวก-ลบ หลักรงนกพิราบ Pascal identity และ Binomial theorem		✓	△	✓
กราฟไม่มีทิศทาง กราฟมีทิศทาง กราฟถ่วงน้ำหนัก degree adjacency และ labels			✓	✓
multigraph และกราฟที่มี self-loop			△	✓
path, cycle, tour, walk, Euler tour, Hamiltonian path/cycle และ reachability			✓	✓
connected component, shortest distance, biconnected graph และ strongly connected graph			✓	✓
tree, rooted tree, binary tree, forest, spanning tree และ subgraph			✓	✓
traversal strategies, bipartite graph, DAG, complete graph และ star			✓	✓
center/centroid ของ tree และสมบัติพื้นฐานเชิงจัดหมู่ของกราฟ เช่น handshaking lemma			△	✓
planar graph และ Euler characteristic เบื้องต้น				✓
6.1 — Programming Fundamentals	✓	✓	✓	✓
องค์ประกอบของคอมพิวเตอร์ โปรแกรมคอมพิวเตอร์ และประเภทของภาษาคอมพิวเตอร์				
โครงสร้างภาษาและการเขียนโปรแกรมด้วย Python	✓			
โครงสร้างภาษาและการเขียนโปรแกรมด้วย C/C++	✓	✓	✓	✓
syntax และ semantics พื้นฐานของภาษาโปรแกรมระดับสูงอย่างน้อยหนึ่งภาษา	✓	✓	✓	✓
ข้อมูล ชนิดข้อมูล ตัวแปร ค่าคงที่ ตัวดำเนินการ นิพจน์ assignment และคำสั่ง	✓	✓	✓	✓

ต่อหน้าถัดไป

ตารางที่ 3: เนื้อหาการอบรมวิชาคอมพิวเตอร์ (ต่อ)

เนื้อหา	TOI-Zero	ค่าย 1	ค่าย 2	TOI/IOI
primitive types: boolean, integer, character	✓	✓	✓	✓
simple I/O การรับข้อมูลนำเข้า และการแสดงผลข้อมูลส่งออก	✓	✓	✓	✓
โครงสร้างควบคุมของโปรแกรม เช่น if/else, while และ for	✓	✓	✓	✓
การเขียนโปรแกรมแบบมีโครงสร้าง การวางแผนโปรแกรม การออกแบบด้วยผังงาน และรหัสเทียม	✓	✓	✓	✓
ฟังก์ชัน parameter passing การคืนค่าจากฟังก์ชัน และ structured decomposition		✓	✓	✓
ขอบเขตของตัวแปร การแยกงานที่ทำซ้ำ และการจัดโครงสร้างโปรแกรม		✓	✓	✓
recursion: แนวคิด ฟังก์ชันเรียกซ้ำ mutual recursion และการ implement recursion		△	✓	✓
recursive backtracking และ divide-and-conquer strategies			✓	✓
พอยน์เตอร์ references คลาส และ struct		✓	✓	✓
การใช้ real numbers ในโจทย์ที่เสถียรเชิงตัวเลข และ precision issues			△	✓
6.1 PF3 / 6.2 AL3b — Fundamental Data Structures	✓	✓	✓	✓
อาร์เรย์หนึ่งมิติ และการประมวลผลรายการข้อมูลพื้นฐาน				
อาร์เรย์หลายมิติ	✓	✓	✓	✓
strings และ string processing พื้นฐาน		✓	✓	✓
static allocation, stack allocation และ linked structures		✓	✓	✓
กองซ้อนและแถวคอย (stack and queue)		✓	✓	✓
รายการโยง (linked list)		✓	✓	✓
การแทนกราฟและต้นไม้ เช่น adjacency list และ adjacency matrix			✓	✓
กลยุทธ์เลือกใช้โครงสร้างข้อมูลที่เหมาะสม			✓	✓
การใช้โครงสร้างข้อมูลแบบ associative เช่น map และ unordered_map			✓	✓
การใช้แถวคอยลำดับความสำคัญ (priority queue) และ binary heap			✓	✓
disjoint set union / Union-Find			✓	✓
Fenwick tree และ segment tree			△	✓
tries			△	✓
balanced/augmented binary search trees			△	✓
LCA สำหรับ static rooted tree			△	✓
heavy-light decomposition และ centroid decomposition สำหรับ static trees				✓
persistent data structures แบบพื้นฐาน				✓

ต่อหน้าถัดไป

ตารางที่ 3: เนื้อหาการอบรมวิชาคอมพิวเตอร์ (ต่อ)

เนื้อหา	TOI-Zero	ค่าย 1	ค่าย 2	TOI/IOI
nesting data structures เช่น sequence of sets				✓
6.1 PF2 / 6.2 – Problem Solving, Algorithms, and Complexity		✓	✓	✓
ตัวอย่างโจทย์และการทดลองแก้โจทย์ด้วยการเขียนโปรแกรม				
กลยุทธ์การแก้ปัญหา: understand-plan-do-check, separation of concerns, generalization, specialization, case distinction, working backwards	✓	✓	✓	✓
บทบาทของอัลกอริทึม กลยุทธ์การ implement และกลยุทธ์การ debug ในกระบวนการแก้ปัญหา		✓	✓	✓
algorithm specification, precondition, postcondition, correctness และ invariants			△	✓
การวิเคราะห์ประสิทธิภาพเบื้องต้น asymptotic analysis และ Big O notation			✓	✓
standard complexity classes และ time-space tradeoffs			△	✓
empirical performance measurements			△	✓
simple loop design strategies และ brute force / exhaustive search		✓	✓	✓
การเรียงข้อมูล และการเรียงลำดับอย่างง่าย เช่น bubble/selection/insertion sort		✓	✓	✓
การประมวลผลอาร์เรย์พื้นฐาน เช่น อ่านข้อมูลซ้ำ นับ รวม ค่าเฉลี่ย และหา min/max	✓	✓	✓	✓
simple array manipulation เช่น fill, shift, rotate, reverse, prefix sums, histogram, bucket sort		✓	✓	✓
การค้นหาข้อมูลแบบลำดับ simple string algorithms และ naive substring search		✓	✓	✓
binary search และ sequential search		✓	✓	✓
simple integer algorithms: radix conversion, Euclid algorithm, primality test, sieve, factorization, fast exponentiation			✓	✓
quicksort, quickselect, heap sort และ merge sort			△	✓
tree traversals: preorder, inorder, postorder, DFS และ BFS			✓	✓
applications of DFS tree เช่น topological ordering, Euler paths/cycles, connected components และ transitive closures			✓	✓
shortest path algorithms: Dijkstra, Bellman-Ford, Floyd-Warshall			✓	✓
minimum spanning tree: Prim/Jarnik และ Kruskal			✓	✓
bridges, articulation points และ strongly connected components			△	✓
greedy algorithms, divide and conquer, backtracking, branch-and-bound และ dynamic programming			✓	✓
simple arbitrary precision integer operations			△	✓

ต่อหน้าถัดไป

ตารางที่ 3: เนื้อหาการอบรมวิชาคอมพิวเตอร์ (ต่อ)

เนื้อหา	TOI-Zero	ค่าย 1	ค่าย 2	TOI/IOI
maximum bipartite matching และ maximum network flow			△	✓
combinatorial game basics, winning/losing positions และ minimax				✓
simple grammar in Backus-Naur Form			△	✓
amortized analysis เบื้องต้น			△	✓
6.2 AL10 – Geometric Algorithms			△	✓
representing points, vectors, lines และ line segments				
area of polygon from coordinates และ coordinate compression			△	✓
collinearity, parallel/orthogonal vectors และ clockwise turns ด้วย dot/cross products			△	✓
intersection of two lines และ point in polygon / convex polygon			△	✓
O(n log n) convex hull และ sweeping line method				✓
7 SE3 / 8 – Development Tools and Environments		✓	✓	✓
การเริ่มต้นเขียนโปรแกรม การใช้งานคอมพิวเตอร์ และเครื่องมือสำหรับพัฒนาโปรแกรม เช่น IDE หรือตัวแก้ไขโค้ด				
การใช้ไลบรารีมาตรฐานของ C++ (C++ STL)		✓	✓	✓
write/edit code, compile and execute programs, debug own programs		✓	✓	✓
use GUI, OS, supporting applications, program development tools, web browser and file management	✓	✓	✓	✓
TOI-Zero / 7 SE6 – Grader and Validation	✓			
การใช้งาน TOI National Grader: การสมัคร การเข้าสู่ระบบ การเลือกโจทย์ การส่งคำตอบ และการอ่านผลตรวจ				
การแปลผลจาก Grader และการปรับแก้โปรแกรมจากผลตรวจ	✓			
การเขียนโปรแกรมเพื่อแก้ปัญหา การตรวจสอบโปรแกรม การวัดผล การใช้ระบบตรวจอัตโนมัติ และสภาพแวดล้อมของการแข่งขัน		✓	✓	✓
testing fundamentals, test plan and test case generation		✓	✓	✓
black-box/white-box testing, unit/integration/validation/system testing				✓
inspections / code review / built-in tests		✓	✓	✓
7 – Software Engineering Practices		✓	✓	✓
fundamental design concepts, structured design และ transform abstract algorithm into efficient program				
read/write text files in prescribed simple format		✓	✓	✓

คำอธิบายท้ายตาราง:

- กลุ่มอ้างอิงหลัก: 5 = Mathematics, 6 = Computing Science, 7 = Software Engineering, 8 = Computer Literacy

- TOI-Zero ใช้หัวข้ออาร์เรย์ในระดับพื้นฐาน เช่น เก็บข้อมูลชุดเดียว วนอ่านค่า นับ รวม และหา min/max
- หัวข้อ เมทริกซ์เบื้องต้น เป็นบริบทเสริม/พิจารณาตัด เนื่องจากไม่อยู่ในขอบเขต IOI Syllabus 2025

เรื่องที่จะไม่สอน

หัวข้อต่อไปนี้ไม่บรรจุเป็นเนื้อหาสอนในหลักสูตรสาขาคอมพิวเตอร์ ปี 2569 แต่อาจปรากฏในบริบทอื่นของ IOI Syllabus 2025 หรือการศึกษาต่อระดับสูง

ตารางที่ 4: เรื่องที่จะไม่สอน

เรื่องที่จะไม่สอน
5.1 Arithmetic and Geometry เรขาคณิตสามมิติหรือมิติสูงกว่า การวิเคราะห์และเพิ่มความแม่นยำของการคำนวณ floating-point การหารมอดุลาร์และอินเวอร์สเชิงคูณ จำนวนเชิงซ้อน ตรีโกณมิติ ภาคตัดกรวยทั่วไป เช่น พาราโบลา ไฮเพอร์โบลา และวงรี
5.2 Discrete Structures คาร์ดินาลิตี้และการนับได้ของเซตอนันต์ normal forms validity และข้อจำกัดของ predicate logic การแก้ recurrence relations Burnside lemma hypergraph, perfect graph, treewidth/expansion, planarity testing, planar separator ความน่าจะเป็นแบบไม่ต่อเนื่องในกรณี finite
5.3 Other Mathematics linear algebra ขั้นสูง เช่น matrix multiplication/exponentiation/inversion, Gaussian elimination, polynomial interpolation, FFT calculus combinatorial game theory ขั้นสูง เช่น Sprague-Grundy statistics
6.1 Programming Fundamentals data representation in memory, heap allocation, runtime storage management, exact fractions
6.2 Algorithms and Complexity best/average/worst case, little-o/Omega/Theta, tuning parameters, average-case analysis, recurrence analysis heuristics, approximation, randomized algorithms, ML feature finding matroid-based optimization, lexicographic BFS, advanced matching/flow, alpha-beta pruning

ต่อหน้าถัดไป

ตารางที่ 4: เรื่องที่จะไม่สอน (ต่อ)

เรื่องที่จะไม่สอน

KMP, Rabin-Karp hashing, suffix arrays/trees/automata, Aho-Corasick

complex heaps, hash tables, 2D tree-like structures, advanced persistence, dynamic tree data structures

distributed algorithms

computability: tractable/intractable, uncomputable functions, halting problem

P/NP, nondeterminism, NP-hardness reductions

finite-state machines, context-free grammars, regular expressions

online/randomized algorithm analysis

cryptographic algorithms

point-line duality, halfspace intersection, Voronoi, Delaunay, circle intersections, high-dimensional LP, center of mass, transformation composition

parallel algorithms

6.3 Other Areas in Computing Science

architecture/organization, OS, cloud, programming-language theory, HCI, intelligent systems, information management, social issues, computational science

basic processing of graphical data

graphics libraries เช่น OpenGL

7-8 Software Engineering and Computer Literacy

software architecture, design for reuse, OO analysis/design, component-level design

programming by example, API debugging environment, class browsers, component-based intro

testing/configuration/requirements/design modeling tools and tool integration

process assessment models and software metrics

prototyping and requirements analysis modeling

validation planning and object-oriented testing

software maintenance, maintainability, re-engineering, legacy systems, reuse

SQA, team management, measurement/estimation, project management tools

component-based computing

formal verification/specification languages/executable specifications

software reliability and specialized systems development

calculator, word processor, spreadsheet, DBMS, e-mail, graphics tools