

SO 1 An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics

PI 1.1 Identify and formulate engineering problems

PI 1.2 Solve engineering problems by applying mathematics and engineering knowledge

Passing criterion $\geq 70\%$

PI 1.1 Identify and formulate engineering problems

Assessment 1.1.1 Midterm exam : identify key properties of data structures that affect the program efficiency

Attainability = 26 out of 37 students (70%)

3.8 มีต้นไม้ 2 ต้นคือ binary search tree ธรรมดา กับ AVL tree ทั้งสองต้นเก็บข้อมูลชุดเดียวกัน จำนวน n ตัว ในกรณีเลวร้ายที่สุด asymptotic runtime ของการค้นหาข้อมูลในต้นไม้ทั้งสองต้นจะเป็นเท่าไร ยกตัวอย่างกรณีเลวร้ายที่สุดมาให้ดูด้วย (4 คะแนน)

ค้นหาโหนดที่มีค่ามากที่สุดหรือน้อยที่สุดในต้นไม้

- BST : โหนดในต้นไม้ทั่วไปฝั่งใดฝั่งหนึ่ง (วาดรูป / อธิบาย) asymptotic runtime = $O(n)$
- AVL : ต้นไม้มีการปรับ balance ทำให้ความสูงไม่มีทางเกิน $O(\log n)$
asymptotic runtime = $O(\log n)$

ABET : Course objective 1, SO 1 → pass at 2 points

Assessment 1.1.2 Midterm exam : formulate runtimes from given code
(i.e. code → set of $O(1)$ operations → runtime complexity)

Attainability = 25 out of 37 students (68%)

หน้า 3 เลขประจำตัว

2.2 จากโปรแกรม 2 โปรแกรมข้างล่างนี้ โปรแกรมแรกใช้ ArrayList เก็บข้อมูล n ตัว โปรแกรมที่สองใช้ LinkedList เก็บข้อมูล n ตัว ถ้า n มีขนาดใหญ่มาก อธิบายการทำงานของแต่ละโปรแกรมว่าทำไมจึงทำงานได้เร็วกว่าหรือช้ากว่าอีกโปรแกรมหนึ่ง (3 คะแนน)

โปรแกรมที่ 1	โปรแกรมที่ 2
<pre>ArrayList<String> A = new ArrayList<String>(); for (int i=0; i < n; i++) { String s = fileScan.next(); A.add(0,s); }</pre>	<pre>LinkedList<String> L = new LinkedList<String>(); for (int i=0; i < n; i++) { String s = fileScan.next(); L.add(0,s); }</pre>
โปรแกรมที่เร็วกว่าคือ โปรแกรม 2	เพราะ LinkedList มีโครงสร้างข้อมูลที่โยงถึงกันด้วย pointer การเพิ่มข้อมูล (prepend) เพียงแค่ปรับการโยง pointer กรณีแย่ที่สุดก็จะได้ runtime = $O(1)$
โปรแกรมที่ช้ากว่าคือ โปรแกรม 1	เพราะ ArrayList มีโครงสร้างเป็นอาร์เรย์ การเพิ่มข้อมูล (prepend) ต้องเลื่อนตัวอื่นๆ ไปข้างหลังก่อน กรณีแย่ที่สุดจะได้ runtime = $O(n)$

ABET : Course objective 2, SO 1 → pass at 1.5 points

Assessment 1.1.3 Final exam : compare the efficiency of different alternative algorithms
 Attainability = 28 out of 37 students (76%)

2.1 การทำงานของ mergesort ตามที่เรียนมา ตอนแบ่งข้อมูลเราแบ่งทีละ 2 ส่วนเท่าๆ กัน (ในสัปดาห์ 1 : 1) สมมติว่าเราเปลี่ยนเป็นแบ่งข้อมูลทีละ 3 ส่วน ในสัปดาห์ 1 : 2 : 3 การแบ่งข้อมูลแบบใหม่จะทำให้การทำงานของ mergesort เมื่อข้อมูลมีปริมาณมากๆ แตกต่างไปจากเดิมอย่างไร ให้แสดงการวิเคราะห์ asymptotic runtime ของ mergesort ของเดิมและของใหม่ มาด้วย (6 คะแนน)

- วิเคราะห์ runtime ของเดิม $O(n \log n)$ 2
- วิเคราะห์ runtime ของใหม่ $O(n \log n)$, อธิบายส่วนที่แบ่งข้อมูลและ merge ทำไมแต่ละครั้งยังได้ $O(n)$ 3
- สรุปว่าไม่แตกต่าง เมื่อ n มีปริมาณมากๆ 1

ABET : Course objective 2, SO 1 → pass at 3 points

Average attainability = $(70 + 68 + 76)/3 = 71.33\%$

Student performance

	Student ID	Assessment 1.1.1 Pass at 2 points	Assessment 1.1.2 Pass at 1.5 points	Assessment 1.1.3 Pass at 3 points
1	6113054	2.5	3.0	5.0
2	6113055	2.0	1.5	0.0
3	6113057	2.5	2.5	5.0
4	6113058	1.5	0.0	2.0
5	6113059	2.5	1.0	4.0
6	6113060	0.0	2.0	3.0
7	6113062	0.0	2.5	3.0
8	6113063	2.5	2.5	2.5
9	6113064	4.0	0.0	4.5
10	6113065	1.0	2.0	2.5
11	6113127	2.5	1.0	5.0
12	6113128	0.5	1.5	1.0
13	6113129	2.0	2.0	3.0
14	6113130	2.0	0.0	4.0
15	6113131	2.0	1.0	4.0
16	6113132	3.0	2.0	0.0
17	6113135	3.0	3.0	4.0
18	6113136	0.5	1.5	5.0
19	6113137	2.0	2.0	3.5
20	6113138	4.0	3.0	4.5
21	6113139	2.0	3.0	3.0
22	6113140	4.0	2.0	5.0
23	6113142	3.0	0.0	3.0
24	6113143	0.5	1.5	0.0
25	6113187	4.0	1.0	3.0
26	6113188	1.5	1.5	5.0
27	6113223	3.0	0.0	1.0
28	6113224	3.5	3.0	4.0
29	6113225	2.0	2.0	3.0
30	6113226	2.0	2.0	4.0
31	6113227	2.5	3.0	5.0
32	6113228	1.0	0.0	3.0
33	6113229	2.0	3.0	5.0
34	6113293	1.5	2.5	3.0
35	6113294	0.5	1.0	2.0
36	6113295	3.0	1.0	3.0
37	6113296	4.0	2.5	5.0

PI 1.2 Solve engineering problems by applying mathematical and engineering knowledge

Assessment 1.2.1 Group project 2. Implement Java program to solve knight path puzzle and write a report explaining data structures and algorithms chosen for the project (pass at 10 out of 15 points)

Attainability = 37 out of 37 students (100%)

Average attainability = 100%

Student performance

	Student ID	Assessment 1.2.1 Pass at 10 points
1	6113054	14.5
2	6113055	14.5
3	6113057	14.0
4	6113058	13.0
5	6113059	13.5
6	6113060	13.0
7	6113062	13.0
8	6113063	11.0
9	6113064	14.5
10	6113065	14.5
11	6113127	14.0
12	6113128	15.0
13	6113129	14.5
14	6113130	13.5
15	6113131	14.5
16	6113132	11.0
17	6113135	15.0
18	6113136	14.0
19	6113137	13.0
20	6113138	13.5
21	6113139	11.0
22	6113140	14.5
23	6113142	13.0
24	6113143	14.5
25	6113187	10.0
26	6113188	15.0
27	6113223	14.5
28	6113224	14.5
29	6113225	10.0
30	6113226	13.0
31	6113227	14.0
32	6113228	10.0
33	6113229	14.5
34	6113293	10.0
35	6113294	13.5
36	6113295	15.0
37	6113296	14.5

Summary

	PI	Attainable ($\geq 70\%$)	Reasons	Remedial Actions	Action Plan	Measurements
SO1	1.1	Yes	<ul style="list-style-type: none">Students could use Java APIs, but some could not tell how different API classes affect the program performance	<ul style="list-style-type: none">More examples / case studies with large input	Next year	Next year
	1.2	Yes				