An Experience of Teaching Computer Science for University Students with Computer Science Unplugged

Ben Tsutom WADA 和田勉[†1]

Abstract: I teach university(undergraduate) class "Algorithm and Data Structure" partially using Computer Science Unplugged activities. In this presentation I introduce the configuration of this class including the usage of some CS Unplugged activities, the score of students, the result of the questionnaire from students, and show the possibility to apply CS Unplugged to education in universities.

Keywords: Computer Science Unplugged, education of informatics in universities

1. Introduction

Computer Science Unplugged[1][2] is a collection of methods for teaching computing science. Although it was originally designed for teaching children such as primary school pupils, but there are some trials to apply them for teaching students in universities[3]. I also use it widely in my class “Algorithms and Data Structure” for recent 10 years.

I have already reported about it twice [4][5]. In this presentation, I introduce the configuration of this class, and show the result of the questionnaire I have taken for recent 5 years in the class.

2. The Class “Algorithms and Data Structure”

The recent (2017) configuration of the class is shown in Table 1. This class consists of 30 school hours, each of which is 90 minutes long. As two consecutive hours are held each week, it takes 15 weeks from April to early August. It contains CS unplugged activities, some other activities similar to CS unplugged, and some other lectures in traditional style with a blackboard.

Undergraduate 2nd and higher grade students can take this class, which is not mandatory. Our department is not specialized for informatics, but a mixture of informatics, management, and design (such as industrial design.) So there are several kind of students in the class, such as who would like to become a general office worker, who would like to become a industrial designer, and who intend to be a software engineer. So it is not expectable for students to have some common motivation. Although I designed this class as a liberal arts class, but also aim at those students who intend to be a specialist of informatics. I use two textbook [2] and [6] for this class.

In those hours I use some CS Unplugged activity, in many cases I let students do that activity, and later lecture about the theory underlying that activity.

3. Some CS Unplugged activities and an author’s original activity in this class

3.1 Binary numbers (Hour 2)

In the second school hour (which is held immediately after the first hour) I teach about binary numbers. (Fig. 1(a) and 1(b).) At first, students are shown the binary number from 0 to 15 (or 31) using large cards, and then I let students repeat it by themselves, and then lecture that it is called binary number（2 進数、2 進數）, the difference from the usual decimal number,

Fig. 1(a) Activity of binary numbers

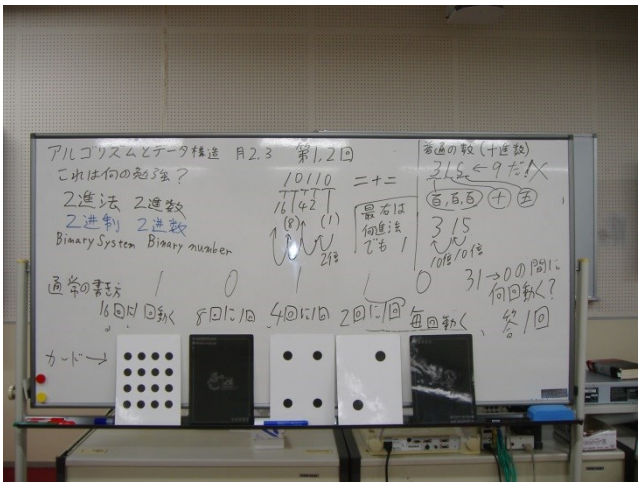

Fig. 1(b) Activity of Binary numbers

†1 Nagano University 長野大學

and its nature.

The characters written in blue color in fig.1(b) is Simplified Chinese(簡體中文). Because there are several foreign students from mainland china, whenever I introduce a technical term, I wrote the word in Chinese, in addition to Japanese and English.

3.2 Battleships - linear search, binary search, hashing (Hour 13-15)

This is an activity for making students understand the concept of several search algorithms. At first students play the Battleship activity, a game of searching a specific "ship" from 26 ships (A ship to Z ship) in linear (or random) search method, binary search method, and hashing (Fig.2(a)). And then the linear search and binary search concept are presented. At that time I emphasize about the large difference at the number of comparisons in linear search (order(n)) and in binary search (order(log n)). For example, if the array to be searched has 1,000,000,000 elements, a specific element can be searched only by watching the array 30 times with binary search method (Fig.2(b)).

3.3 Lightest and Heaviest—Sorting Algorithms (Hour 16-18)

Prepare a balance scale (made of a clothes hanger) and eight cups of different weight whose inside is invisible (made by putting several small glass balls in a paper cup and cover it with a facial tissue) for each student(Fig.3(a)). At first let students sort those eight cups in any manner, which sometimes might be a wrong algorithm. Then I explain the selection-sort algorithm and let students simulate it, and explain about the number of

Table 1. Time allocation (syllabus)

Class "Algorithms and Data Structure," Apr.-Aug.,2017
30 school hours (a school hour consists of 90 mins) and a final examination.
Two consecutive school hours (i.e. consecutive 90 mins – 10 mins break - 90 mins.) are held each week.
This is the actual time allocation. The syllabus planned in advance was slightly different.

School Hour	Contents of the school hour	Corresponding CS Unplugged activity[2] and other similar activities. *:Not a CS Unplugged Activity
1, 2	Introduction Binary numbers, Run-length coding	Count the Dots — Binary Numbers Colour by Numbers — Image Representation
3, 4	ALGO-LOGIC 2 (a website for exercising algorithmic method.) Binary number system	ALGO-LOGIC 2*[7]
5, 6	Binary number system, Character code, Digitizing analog signal	
7, 8	Digitizing of images and movies Data compression – lossless compression and lossy compression	You Can Say That Again! —Text Compression ("Pitter Patter" and "Banana")
9, 10	What is algorithm? Refinement of algorithm Three elements of algorithms - sequence and selection	ALGO-LOGIC 2*
11, 12	Three elements of algorithms - selection and iteration	ALGO-LOGIC 2*
13, 14	Search – linear search and binary search	Battleships—Searching Algorithms
15, 16	Search – hashing Sorting – selection sort	Battleships—Searching Algorithms Lightest and Heaviest—Sorting Algorithms
17, 18	Order of computing Sorting – bubblesort, insertion sort, quicksort	Lightest and Heaviest—Sorting Algorithms Sorting Algorithms Animations*[8]
19, 20	Examination (first) Sorting – mergesort - recursion	
21, 22	Sorting – mergesort – the number of comparison Limit of computer speed: the light speed – parallel computing Module and argument	 SCRATCH*[9]
23, 24	Recursion – factorial, Fibonacci number, reverse of string Towers of Hanoi	Original presentation (see 3.4)
25, 26	Towers of Hanoi – algorithm by recursion Song "Sokkuri House (Identical House)"[10] Increase by exponential order Semi-log graph	Original presentation (see section 3.4)
27, 28	Logarithm Minimum spanning trees, Kruskal's algorithm Packet, Deadlock	The Muddy City—Minimal Spanning Trees The Orange Game—Routing and Deadlock in Networks (see Fig.5)
29, 30	Data structure – record, array, queue, stack, tree Sort by binary tree	
Exam.	Examination (final)	

Fig. 2(a) Activity of "Warship"

comparisons according to the number of cups. Next, I explain about bubblesort, insertion sort, and, as a better way I introduce quicksort (Fig.3(b)) and let students simulate each and think about the number of comparisons of each algorithm.

3.4 Towers of Hanoi – my original explanation (Hour 24-26)

Tower of Hanoi is widely used as an interesting puzzle representing the concept of recursion. In this class I explain the concept of recursion with it in my original manner.

It is widely said that many students give up understanding computing science at the point they study the concept of recursion. In most popular textbooks the recursion (recursive

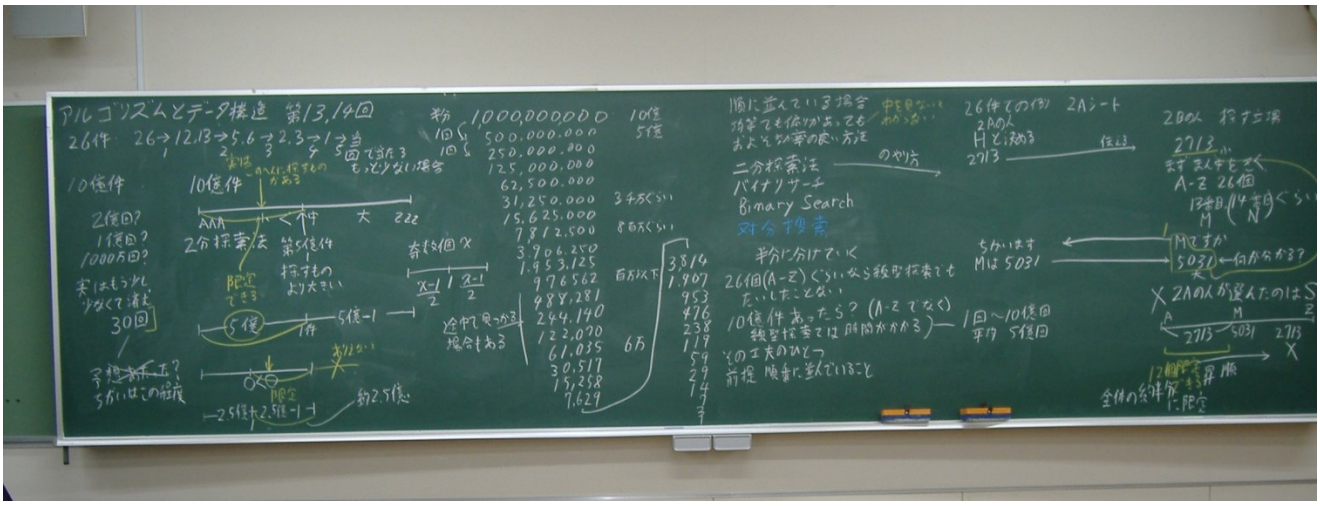

Fig. 2(b) Activity of "Warship"
(Explaining that an array of 1 billion elements- long can be searched only in 30 times with binary search method.)

Fig. 3(a) "Lightest and Heaviest—Sorting Algorithms"

call) is defined as "to call (the procedure) itself," which makes confusing many novice students, because they misunderstand "a procedure calls that procedure itself" as if it is just that "it jumps to the beginning of that procedure." Of course it is not correct. They do not understand that, although the calling and called procedure is identical, the executing point (i.e. the "program counter") and the value of those arguments are independent each other. So I stop telling students "the procedure calls itself" and try to explain that "the procedure calls another copy of itself." I prepare many copies of the procedure printed largely on paper, and use each copy for each call (Fig.4(a) and (b).) Although it is not an almighty way for making students understand the concept of recursion, but some students understand recursion in

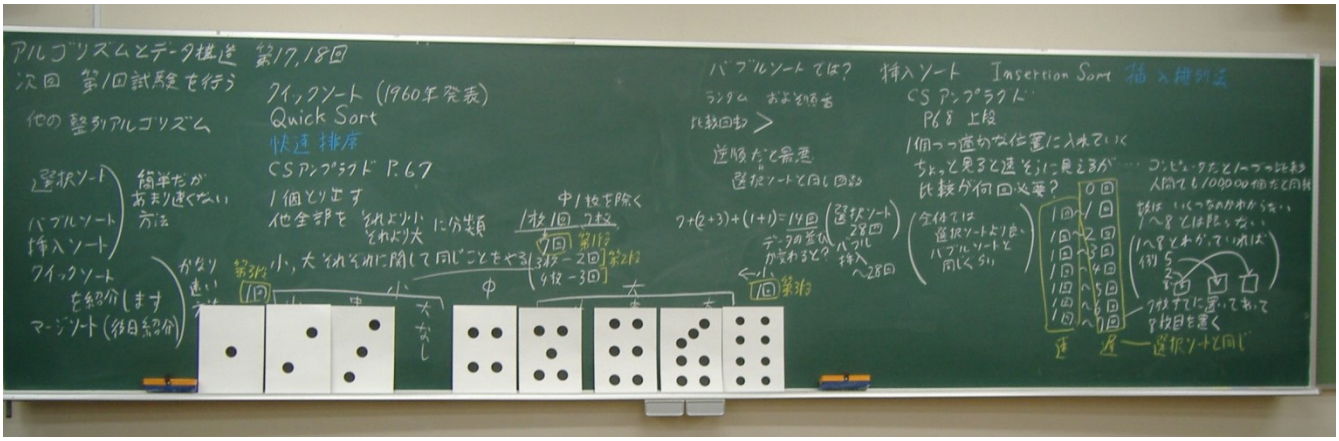

Fig. 3(b) "Lightest and Heaviest—Sorting Algorithms"

Fig. 4(a) Towers of Hanoi

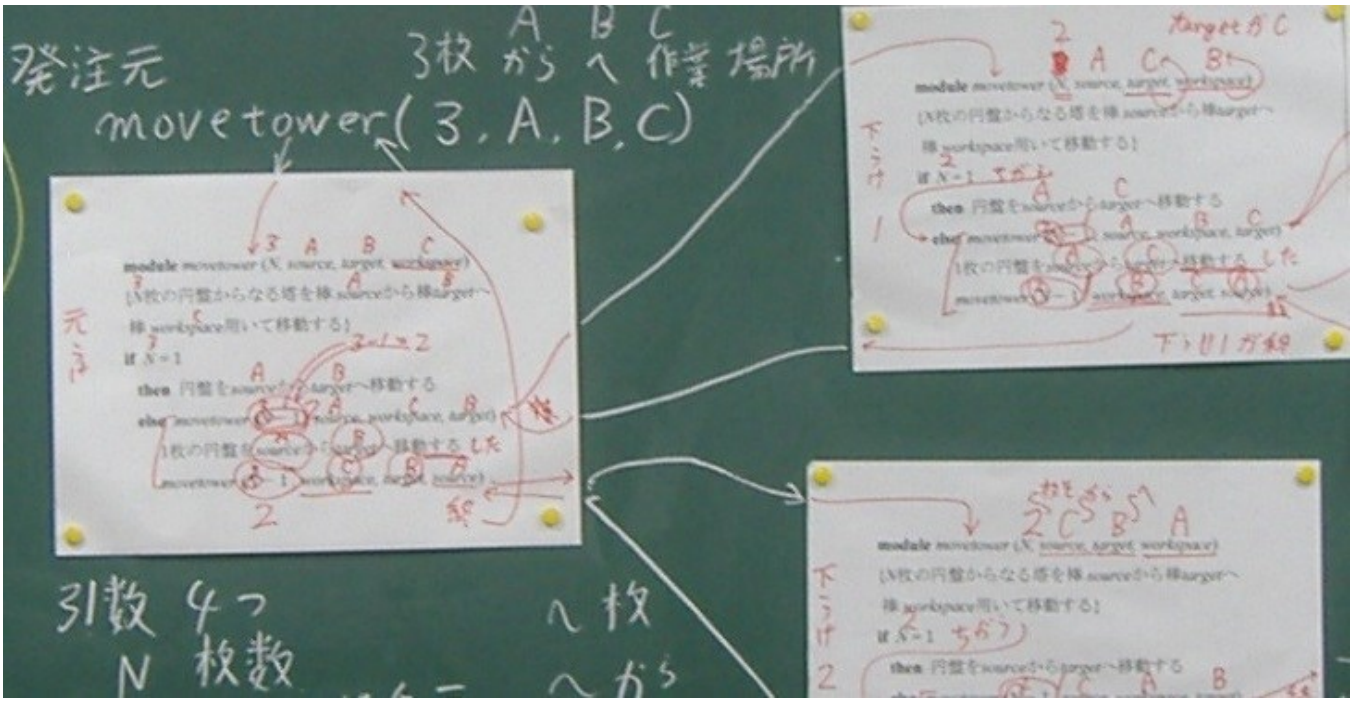

Fig. 4(b) Towers of Hanoi – a part of Fig.4(a) enlarged

this way, that is, "calling the procedure of the name same as itself" means "there are many copies of an identical procedure inside and they are called one by one.

4. Students' score and questionnaire result

The score of students for 5 years (2013-2017) is shown in Table 2. And the result of questionnaire are shown in Table 3. Each year I executed paper tests twice. I did one in the middle of whole class (19th school hour in 2017,) and the other after completing the whole hours. At last I gave each students the point p=min((a + b)/2, b), where a is the score of that student at the first test, and b is that of the final one (each is out of 100.) Table 2 shows the distribution of this p, excluding those students who attended only 19 school hours or less out of 30 hours **and** got only 59 point or less out of 100, who are decided not to be a subject for giving any point.

Unfortunately the result shown in Table 2 and 3 are not connected, that is, it is unknown that which answer sheet is

Table 2. Students' score by school year

Year	2013	2014	2015	2016	2017
秀 Excellent 100～90 points	3	1	2	3	4
優 Good 89～80 points	3	2	3	4	6
良 Middle 79～70 points	5	6	2	2	6
可 Acceptable 69～60 points	2	3	2	3	2
不可 Not acceptable 59points～	11	2	2	2	9

Table 3. Result of Questionnaire by school year

	Ave.	2013	2014	2015	2016	2017
Run-length coding	4.24	4.0	4.2	4.5	4.2	4.4
Minimum spanning trees	4.17	4.0	4.4	4.2	4.7	3.9
Binary numbers	4.17	4.1	4.3	4.5	4.1	4.0
Orange game	4.14	4.1	4.4	3.9	4.6	4.0
Game of hashtable*	3.93	-	3.9	-		-
Towers of Hanoi*	3.93	3.8	3.8	4.2	4.5	3.7
Text Compression "Banana"	3.90	4.1	3.2	4.1	4.0	4.0
Battleships—Searching Algorithms	3.90	4.2	3.7	4.1	3.7	3.8
ALGO-LOGIC*	3.88	3.7	3.8	4.2	4.0	3.9
Beat the Clock —Sorting Networks	3.86	3.5	4.1	3.4	4.6	-
Text Compression "Pitter Patter"	3.82	3.9	3.7	4.3	3.7	3.8
Song "Sokkuri House (Identical House)"*	3.72	3.7	3.8	3.4	4.2	3.5
Lightest and Heaviest —Sorting Algorithms	3.71	3.8	3.6	4.3	3.6	3.6
Increase in exponential order*	3.61	-	-	3.9	3.9	3.4
Recursion – factorial, reverse of strings*	3.51	3.6	3.5	3.1	3.9	3.3
Semi-log graph*	3.43	-	-	3.6	3.8	3.2
Sorting Algorithms*	3.38	-	-	-	3.4	-

*: not included in the original CS Unplugged activities.
-: not done in the school year.
Performed activities and their order are slightly different by school year.
Average means the weighted mean, that is, (the sum of those points the activity got in 5 years) / (the number of answers for the question in 5 years (excluding invalid answers.))

written by, for example, a student who got Excellent(秀) score. "Run-length coding" got the best score in the average for 5 years, followed by "Minimum spanning tree" and the "Binary numbers" and then by "Orange game." Although the worst one was "sorting algorithms," it was held only in 2016. (Corresponding contents are included in "Lightest and Heaviest" in other school years.) Among those activities done for 3 years or more, "Semi-log graph," got the worst score, and "Recursion – factorial, reverse of strings" followed. But these are not original CS Unplugged activities. Among original CS Unplugged activities, "Lightest and Heaviest－Sorting Algorithms" was the worst one.

Those activities which got low scores, such as Semi-log graph, Recursion, Increase in exponential order, Lightest and Heaviest — Sorting Algorithms, seem to be relatively "difficult" ones for our students. Students cannot understand them only with their "common sense" they already have in their daily life, but they must acquire some conceptual matter newly into their mind with much intelligent effort. For example, the exponential order, in which 1,2,4,8... is followed by 1,048,576 only 20 elements after, is much different from their usual intuitive common sense. The log-scale is so too, in which 1,2,3... is followed by 10,20,30... and then by 100,200,300..., in contrast that 1,2,3,...10 must be followed by 11,12,...20 in their natural common sense.

So it might be unavoidable that, in some extent, these activities can get only small scores in the questionnaire. But the activity of Binary numbers, which is not usual in their daily life in which only decimal numbers are used, got relatively high score in the questionnaire. The reason might be that they can get the answer easily by counting "the total number of dots" without thinking or understanding the "difficult" concept.

5. Conclusion

I teach this class which include CS unplugged activities from 2008, and let students answer the questionnaire from 2013. From the next year I would like to perform a questionnaire which can be connected to the score of the student who write it.

Fig. 5 "The Orange Game—Routing and Deadlock in Networks"

References

[1] ”CS UNPLUGGED Computer Science without a computer”. http://csunplugged.org/ (accessed 2017-08-26).
[2] Tim Bell, Ian H. Witten and Mike Fellows, “CS UNPLUGGED An enrichment and extension programme for primary-aged students “, http://csunplugged.org/books/.
Japanese Translation(earlier version) : 兼宗進監訳“コンピュータを使わない情報教育　アンプラグドコンピュータサイエンス、イーテキスト研究所、ISBN-13: 978-4904013007,(2007 年)。
Chinese(Traditional) translation: 翁佳驥等“CS UNPLUGGED 不插電的資訊科學”，中華民國軟體自由協會(2016).
Chinese(Simplified) translation: “CS UNPLUGGED 不插电的计算机科学−课程包”, 华中科技大学出版社，ISBN：978-7-5609 (2010).
[3] Masaru KADA ”UNPLUGGED for college students, as a material for technical English comprehension and mock classes”. 嘉田勝：大学生もアンプラグドー洋書購読と模擬授業による授業実践 Summer Symposium in Samdo SSS2008, D-2, (In Japanese).
[4] Ben Tsutom WADA “An university's class on algorithms using both Computer Science Unplugged and chalkboard lecture”, 和田勉：アンプラグドコンピュータサイエンスと板書講義を併用した大学でのアルゴリズムの授業, IPSJ SIG Technical Report 2009-CE-100(5),pp.1-7 (2009) (In Japanese).
[5] Ben Tsutom WADA “An university's class on algorithms using both Computer Science Unplugged and chalkboard lecture - its improvement thereafter and evaluation from students,コンピュータサイエンスアンプラグドと板書講義を併用した大学でのアルゴリズムの授業ーその後の改良と学生からの評価, IPSJ SIG Technical Report, 2013-CE-121, pp.1-7(2013)(In Japanese).
[6] Les Goldschlager, Andrew Martin Lister ”Computer Science: A Modern Introduction: 2nd edition” Prentice Hall, 1988. (Japanese translation:武市正人他訳：計算機科学入門〔第 2 版〕, サイエンス社（2000 年）)
[7] JEITA (Yutaka OHYAMA) “ALGO-LOGIC”. http://home.jeita.or.jp/is/highschool/algo/index_en.html, (accessed 2017-09-01).
[8] Toptal ”Sorting Algorithms Animation” https://www.toptal.com/developers/sorting-algorithms/, (accessed 2017-09-01).
[9] SCRATCH: https://scratch.mit.edu/
[10] (a music compact disc) Hiroko TANIYAMA “Sokkuri-house”, Yamaha Music Communications, JAN:4542519001469.

Acknowledgments All students gave me the permission to take picture in the class, and answered questionnaire for contributing this research.