Automatic Assessment of Exercises for Algorithms and Data Structures – a Case Study with TRAKLA2

Mikko-Jussi Laakso and Tapio Salakoski
University of Turku
Department of Information Technology
Turku, Finland

Ari Korhonen and Lauri Malmi
Helsinki University of Technology
Department of Computer Science and Engineering
Espoo, Finland

{milaak,tapio.salakoski}@it.utu.fi,{archie,lma}@cs.hut.fi

Abstract

This paper presents the results of the case study introducing TRAKLA2 system in the course of data structures and algorithms at the University of Turku in 2004. We compared students' learning results with the results of the previous year. The numerical course results were clearly better than in 2003 when only pen-and-paper type exercises in classrooms were used. In addition, a survey was made with over 100 students on the changes in their attitudes towards web-based learning environments while getting acquainted with a wholly new system providing them automatic feedback and the option to resubmit their solutions. Our results show that such an on-line learning environment considerably increases positive attitudes towards web-based learning, and according to students' self-evaluations, the best learning results are achieved by combining traditional teaching and www-based learning.

1 Introduction

Automatic assessment (AA) tools for CS courses are being developed and gaining acceptance more and more widely at university level education. The survey of the ITiCSE working group "How shall we assess this" in 2003 indicated clearly that the experience of using AA tools correlates with a positive attitude towards applying such methods more widely, also when assessing higher order skills (Carter et al., 2003). The field where AA is most widely used is assessing programming exercises (*e.g.* Higgins et al. (2002); Luck and Joy (1999); Saikkonen et al. (2001); Vihtonen and Ageenko (2002)). Other applications include grading algorithm exercises (Bridgeman et al., 2000; Hyvönen and Malmi, 1993; Korhonen and Malmi, 2000) and analyzing object-oriented designs and flowcharts (Higgins et al., 2002).

In this paper, we report the experiences on using the TRAKLA2 system for assessing algorithm exercises in which students simulate working of algorithms on a conceptual level. TRAKLA2 by Malmi et al. (2004); Korhonen et al. (2003) is a visual algorithm simulation exercise system that has been developed at Helsinki University of Technology (HUT). Students solve the exercises using graphical manipulation of conceptual visualizations of data structures on the screen. The system provides automatic formative and summative feedback on their work, and allows for resubmitting the solutions.

TRAKLA2 exercises were used for the first time in the basic data structures and algorithms courses at HUT in spring 2003. The system was used in parallel with the old TRAKLA system so that in total 14 TRAKLA2 exercises and 24 TRAKLA exercises were used in two courses[1]. In 2004, only TRAKLA2 was used and the total number of exercises was 26. During these two years more than 1000 students used the system.

In 2004, the University of Turku (UTU) also adopted TRAKLA2 for their data structure course with over 100 students. Compared with HUT this was a major cultural change on

[1] There were two versions of the course, one for CS majors and one for students of other engineering curricula.

the course. In HUT we have used automatically assessed algorithm simulation exercise since 1991 using the older TRAKLA tool, and thus the type of exercises and the culture of using automatic assessment is well-established both for the students and teachers. In UTU, however, no such exercises have been applied, except occasionally as pen-and-paper exercises without any automatic assessment.

In all these courses, both at HUT and UTU, TRAKLA2 exercises were a compulsory part of the course, and grading points achieved from the exercises had an effect on the final grade of the courses, although in slightly different ways. In HUT, TRAKLA2 exercises have an overall effect of 30% of the final course grade, whereas at UTU the TRAKLA2 exercises increased the number of examination points. In both institutes the minimum requirement was achieving at least 50% of the maximum points of the TRAKLA2 exercises.

The structure of the paper is the following. In the next section we give an overview of the TRAKLA2 system. Section 3 presents how the system was used in UTU, and how students attitudes and opinions were surveyed. Section 4 presents the results of the survey and final conclusions are included in Section 5.

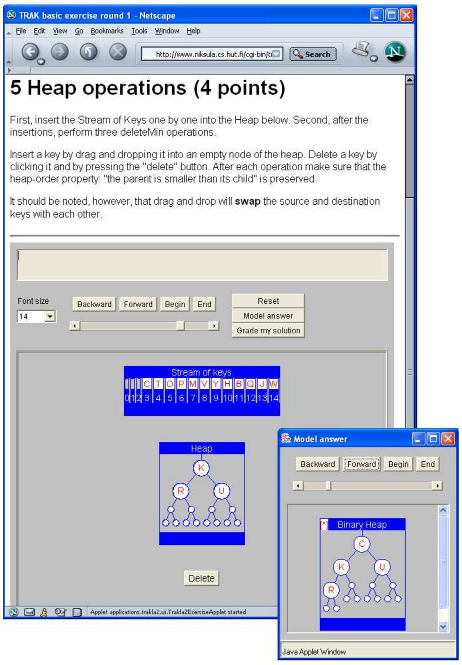
2 Overview of the TRAKLA2 system

TRAKLA2 is a system for automatically assessing *visual algorithm simulation exercises* (Korhonen et al., 2003). It is based on the Matrix algorithm visualization, animation, and simulation framework (Korhonen and Malmi, 2002). TRAKLA2 distributes individually tailored tracing exercises to students and automatically assesses answers to the exercises. In visual algorithm simulation exercises, a learner directly manipulates the visual representation of the underlying data structures to which the algorithm is applied. The learner manipulates these real data structures through GUI operations with the purpose of performing the same changes on the data structures that the real algorithm would do. The answer to an exercise is a sequence of discrete states of data structures resulting from application of the algorithm, and the task is to determine the correct operations that will cause the transitions between each two consecutive states.

Let us consider the exercise in Figure 1. The learner has started to manipulate the visual representation of the Binary Heap data structure by invoking context-sensitive *drag-and-drop operations*. In the next step, for example, he or she can drag the key `C` from a *Stream of keys* into the left subtree of `R` in the binary heap. After that, the new key is sifted up via a *swap* with its parent until the parental dominance requirement is satisfied (the key at each node is smaller than or equal to the keys of its children). The swap operation is performed by dragging and dropping a key in the heap on top of another key. In addition, the exercise applet includes a push button for activating the Delete operation. The `Delete` button is applied in the second phase of the exercise to simulate the deleteMin operation. The student selects a node to be deleted and thereafter uses swap operations to heapify the tree again.

An exercise applet is initialized with proper *randomized input data*. The binary heap exercise, for example, is initialized with 15 alphabetic keys (Stream of keys), that do not contain duplicates. This means that the exercise can be initialized in more than 10^{19} different ways. The learner can *reset the exercise* by pressing the `Reset` button at any time. As a result the exercise is reinitialized with new random keys.

After attempting to solve the exercise, the learner can *review the answer* step by step using the `Backward` and `Forward` buttons. Moreover, the learner can *ask feedback* on his or her solution by pressing the `Grade` button in which case the learner's answer is checked and immediate feedback is delivered. The feedback reports the number of correct steps out of the total number of required steps in the exercise. Finally, it is possible for the student to *submit the answer* to the course database using the `Submit` button. By default an answer to an exercise can be submitted unlimited times; however, a solution for a specific instance of exercise with certain input data can be submitted only once. In order to resubmit a solution



TRAK basic exercise round 1 - Netscape
File Edit View Go Bookmarks Tools Window Help
http://www.niksula.cs.hut.fi/cgi-bin/t
Search

5 Heap operations (4 points)

First, insert the Stream of Keys one by one into the Heap below. Second, after the insertions, perform three deleteMin operations.

Insert a key by drag and dropping it into an empty node of the heap. Delete a key by clicking it and by pressing the "delete" button. After each operation make sure that the heap-order property: "the parent is smaller than its child" is preserved.

It should be noted, however, that drag and drop will **swap** the source and destination keys with each other.

Figure 1: TRAKLA2 applet page and the model solution window.

to the exercise, the learner has to reset the exercise and start over with new randomized input data. Thus, it is not possible to grade the same solution and improve it arbitrarily before submitting it.

A learner can also *examine the model solution* of an exercise. It is represented as an algorithm animation so that the execution of the algorithm is visualized step by step. In Figure 1, the model solution window is opened in the front. The states of the model solution can be browsed using the `Backward` and `Forward` buttons. For obvious reasons, after opening the model solution for given input data, a student cannot submit a solution until the exercise has been reset and resolved with new random data.

Each TRAKLA2 exercise page (*e.g.*, Fig. 1) consists of a description of the exercise, an interactive Java applet, and links to other pages that introduce the theory and examples of the algorithm in question. The current exercise set covers almost 30 assignments on basic data structures, sorting, searching, hashing, and graph algorithms. Appendix A lists the current exercises in TRAKLA2.

3 Algorithms and data structures course at University of Turku

Algorithms and data structures (DSA-UTU) course at University of Turku included 56 lecture hours, 10 classroom exercises (each 2 hours) and 22 TRAKLA2 exercises in spring 2004. Previous courses were held with 56 lecture hours and 13 classroom exercises (2 hours each). The classroom exercises consist of five single exercises like illustrating exercises, proofing exercises, *etc.* TRAKLA2 exercises, however, are most effective to represent exercises in which the task is to illustrate how a specific algorithm works with given input values. Thus, the number of classroom exercises was cut down after TRAKLA2 was taken in use. In numbers, classroom exercises decreased from 65 to 50. Each TRAKLA2 exercise was given points from one to four. There was a possibility to get in total of 47 TRAKLA2 points in DSA-UTU course. The TRAKLA2 exercises were divided into three rounds by synchronizing the exercises to topics in hand in the DSA-UTU course.

3.1 Grading and requirements of the DSA-UTU course

There were two ways of passing the course. By taking the final examination (0-32 course points) or by taking two midterm-examinations (both 0-16 course points). In either way, student must still fulfill the minimum requirements, which are: i) students must do at least 20 of the 50 classroom exercises, ii) students must get at least 50% of the TRAKLA2 points (maximum 47 points), and iii) students must get at least 20 course points out of the total of 40 course points in share.

It was possible to get 32 course points from the examination(s) and eight course points from TRAKLA2 exercises. Conversion of TRAKLA2 points to course points was linear between the minimum requirements 50% (pass with zero course points) and 100% TRAKLA2 points (8 course points that is 20% of the maximum).

In comparison with earlier DSA-UTU courses, TRAKLA2 exercises replaced one question in the examination or a half of a question in both midterm-examinations. Traditionally one of the five questions in examination has been such an illustrative type of assignment, and this was the very question now replaced by TRAKLA2 exercises.

The final grading of the DSA-UTU course was in scale from one to three with 0.25 steps. By getting 20 course points the student will get lowest grade, which is one. In addition, by doing 60% or 80% of classroom exercises, any student can get an additional + or $\frac{1}{2}$ to his grade, respectively (still requires the student to fulfill the course minimum requirements).

3.2 The setting of the study

The attitudes of the students in UTU where studied using questionaries. Three sets of questionaries where filled by the students during the course. The first questionary at the beginning of the course, the second (Mid) at the first midterm-examination (after the first round of TRAKLA2 exercises), and the third one at the second midterm-examination (after the courses).

The first questionary was aimed to gather information about students' attitudes towards and experiences of www-based materials and tools in earlier courses. Questions also covered students' opinions about how well www-based exercises are suitable in DSA-UTU course (scale in numbers 1-5, 5 is the best). It was also asked how students prefer to do DSA-UTU courses exercises (by www-exercises, by classroom-exercises, or mixed). Students ranked different ways of doing exercises in order from one to three (one is the best, three is the worst) by their own interest. In the same way, the students also self assessed the level of their learning.

There were two main questions of yes-no type in the second questionary. The first question was about the contribution of TRAKLA2 system in the learning of course topics. The second question was about usability of TRAKLA2 and about any problems of using it. Both questions included also possibility of free text comments.

On the third questionary, the questions on the first and second questionaries were repeated. In addition, further comments and suggestions were asked for.

4 Results and discussion

As a whole, the TRAKLA2 system has worked well with surprisingly good results both at HUT and at UTU. In 2004, 30% of the students at HUT achieved the maximum number of points for the 26 exercises, and over 55% achieved 90% of the maximum. Only 15% of the students failed to get the required minimum of 50% of the points; in practice these were students who dropped the whole course early. At UTU the results were even better. The average number of points achieved was 7.34 points out of maximum 8 points.

Students' opinions of the system were determined through a web-based survey at the end of the HUT course in 2003. 364 students answered. 80% of them gave an overall grade of 4 or 5 to the system in scale 0–5, where 5 was the best grade. The system was almost unanimously considered to aid learning and easy to use. In UTU, free feedback from the system was well in line with these observations. In addition, a different questionnaire was carried out which surveyed how students' attitudes towards on-line learning environments was changed when they had used TRAKLA2. This pointed out clearly that the attitudes became more positive.

In the following, we present a more detailed analysis of the results of the survey on the UTU students' opinions and attitudes towards www-based learning. Moreover, the learning results are presented based on students' self evaluation. After that, results derived from course statistics are presented, including the numbers of students failed/passed in total, average grades, attendances in classroom and TRAKLA2 exercises, *etc.* The data is compared with the data from DSA-UTU course in spring 2003, when the course was given by the same lecturer and the classroom exercises were very similar to those in spring 2004.

4.1 The survey results

There were 96 students answers to the first questionary ('Start'), 103 to the second ('Mid'), and 81 to the third questionary ('End'). At the Start and End the students were asked about their opinion on the suitability of www-based exercises for learning data structures and algorithms. The answering alternatives were well (5), quite well (4), neutral (3), quite bad (2), and bad (1). The Start average were quite high, 3.94, and the End average were even higher, 4.84. These results indicate that www-based exercises are very suitable for learning data structures and algorithms. Also the increase of the average during the course is large and therefore it seems that www-based exercises were well accepted and approved even by students without strong positive prejudice.

As to the qualitative analysis, also the free text comments were analyzed. There were a number of answers in which students said that it is much more elegant to do this kind of illustrative type of exercises with TRAKLA2 rather than doing the same in a piece of paper step by step. Also, it was often mentioned that TRAKLA2 exercises concretized the actions and operations of an algorithm. It was also confirmed that the immediate feedback by the TRAKLA2 system helped the students to find the point where they made a mistake and encouraged them to further deepen their understanding of the subject. This is also reflected by course statistics.

In the Mid and End questionaries, the students were asked how TRAKLA2 exercises contributed to their learning. In the Mid, the question was formulated as yes/no-type, and 94% of students answered that TRAKLA2 exercises did aid their learning process. At the End, the students were asked to describe the contribution on a scale from 1 to 5 (5 is the best). The average of the answers was 4.10, and 84% of the students selected 4 or 5, while there were only two answers below 3. This result is well in line with previous results from the study at HUT.

We also asked the students to give their preference on the three ways of doing exercises: traditional classroom exercises, web-based exercises, or mixed (see Figure 2). In the same manner, the students were asked to assess the level of their learning (Figure 3). It can be seen from the answers that the students' attitudes changed positively towards www-based exercises during the course. Students prefer the most to do exercises by combining traditional and www-based exercises even in the starting questionary, and their opinion strengthened during the course so that at the end, nearly three out of four students considered mixed exercises the best. The same happened to the students' self assessment of their learning. The mixed alternative is clearly the most suitable way to learn data structures and algorithms. Furthermore, if the students' were to choose only between traditional and web-based exercises, they would prefer traditional over www-based exercises due to their better contribution to learning. This is very interesting result suggesting that although web-based exercises complement very well traditional classroom exercises, the former can hardly replace the latter in general.

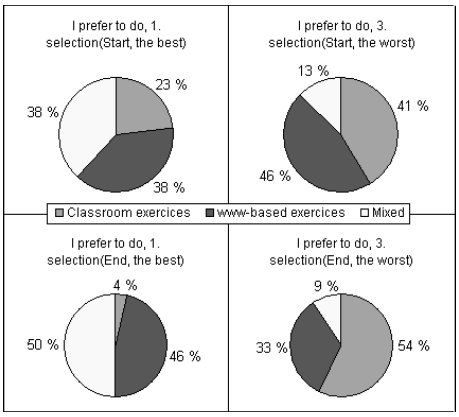


Figure 2: I prefer to do

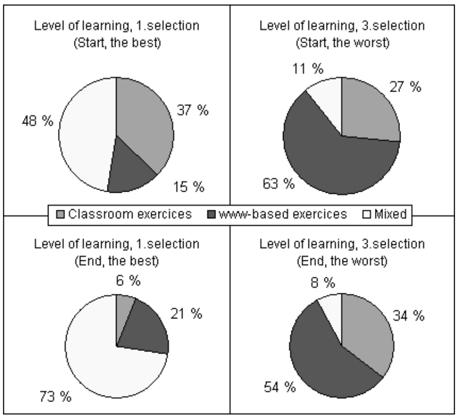


Figure 3: The level of learning

Table 1: Students' activity in classroom exercises

	Spring 2003	Spring 2004
Number of (#) attendants	186	165
Average % of classroom exercises (only who did at least 40%)	54.5	60.3
Number of (#) attendants who did 0% - 40% of classroom exercises(failed)	76	43
# attendants who did 40% - 60% of classroom exercises(no bonus)	80	79
# attendants who did 60% - 80% and received + from classroom exercises to their final grade	18	21
# attendants who did 80% - 100% and received $\frac{1}{2}$ from classroom exercises to their final grade	12	22

4.2 The course statistics

Table 1 shows statistics about students' activity in classroom exercises from DSA-UTU courses in spring 2003 and spring 2004. In addition, students got as an average of 7.34 course points from TRAKLA2 exercises, and 69,2 % of students did 100 % of TRAKLA2 exercises.

As we can see from the statistics, in spring 2004, the students were more active not only in using TRAKLA2 but also in other part of the course compared with 2003; especially, the average performance in classroom exercises raised from 54,5% to 60,3%. There is also a statistically significant difference (χ^2-test, $p < 0.01$) between the two years in the statistics in Table 1. Thus, a larger number of students received additional + / $\frac{1}{2}$ to their final grade in 2004 than in 2003. These observations confirm that the introduction of TRAKLA2 system enhanced the students' motivation and performance on the DSA-UTU course.

In Table 2, there are shown the basic statistics from DSA-UTU courses in 2003 and 2004,

Table 2: Course statistics

	Spring 2003	Spring 2004
Number of (#) attendants	186	165
Average course points	26.15	27.51
Average of the final grades	2.01	1.97
# attendants who were in second midterm-examination	58	82
# passed attendants	49	81
% of attendants who were in second midterm-examination and passed the course	84.5	98.7

which were of about the same size. There was a major increase in number of passed attendants. On the other hand, when looking at the average of course points (t-test, $p = 0.19$) and the average of final grades (χ^2-test, $p = 0.12$), there is no statistically significant difference between those two courses. Combining these two observations it can be concluded that TRAKLA2 aided many students to get over the edge and pass DSA-UTU course. Hence, it seems that TRAKLA2 is truly useful for those students who have difficulties learning data structures and algorithms by classroom exercises.

5 Conclusions

The study showed that students' attitudes strengthened positively towards www-based exercises. Moreover, the mixed alternative is far the most appropriate way to learn topics of DSA course, and it's well approved and preferred by students. Furthermore, the results suggest that www-based exercises constitute a good amendment to DSA course. However, it seems also that there exits a certain desire for more traditional exercises. Whether these students' exceptations can be fulfilled by a future version of TRAKLA2 or similar web based tools, remains an interesting challenge.

Interface of the TRAKLA2 system was easy to use, and features like possibility to get immediate feedback and the resubmit alternative aided students to complete given exercises, and by that they enhanced their learning. In addition, the study pointed out that the TRAKLA2 system affected positively on students' behaviour on other areas of DSA-UTU course, and an average student did more work for learning the course's topics. In the same time, the number of passed attendants raised from 49 to 81, thus the TRAKLA2 system aided especially less talented students to get over the edge and pass the course.

At this time, the only existing type of TRAKLA2 exercise is to illustrate how a specific algorithm works on given input. Basically, this calls for tracing the execution of the algorithm, whereas the system currently offers no support for constructive exercises, such as in which a problem is described, example input and output values are given, and the task is to construct the algorithm.

In conclusion, the TRAKLA2 system was well accepted and approved by students, and it will be used in forthcoming DSA courses also at UTU. A key task of the future is to develop novel types of TRAKLA2 exercises in collaboration between Helsinki University of Technology and University of Turku.

References

Bridgeman, S., Goodrich, M. T., Kobourov, S. G., Tamassia, R., 2000. PILOT: An interactive tool for learning and grading. In: The proceedings of the 31st SIGCSE Technical Symposium on Computer Science Education. ACM, pp. 139–143.
URL `citeseer.nj.nec.com/bridgeman00pilot.html`

Carter, J., English, J., Ala-Mutka, K., Dick, M., Fone, W., Fuller, U., Sheard, J., 2003. ITICSE working group report: How shall we assess this? SIGCSE Bulletin 35 (4), 107–123.

Higgins, C., Symeonidis, P., Tsintsifas, A., 2002. The marking system for CourseMaster. In: Proceedings of the 7th annual conference on Innovation and technology in computer science education. ACM Press, pp. 46–50.

Hyvönen, J., Malmi, L., 1993. TRAKLA – a system for teaching algorithms using email and a graphical editor. In: Proceedings of HYPERMEDIA in Vaasa. pp. 141–147.

Korhonen, A., Malmi, L., 2000. Algorithm simulation with automatic assessment. In: Proceedings of The 5th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education. ACM, Helsinki, Finland, pp. 160–163.

Korhonen, A., Malmi, L., May 2002. Matrix — Concept animation and algorithm simulation system. In: Proceedings of the Working Conference on Advanced Visual Interfaces. ACM, Trento, Italy, pp. 109–114.

Korhonen, A., Malmi, L., Silvasti, P., Nikander, J., Tenhunen, P., Mård, P., Salonen, H., Karavirta, V., 2003. TRAKLA2. URL: http://www.cs.hut.fi/Research/TRAKLA2/ (27.9.2003).

Luck, M., Joy, M., 1999. A secure on-line submission system. Software - Practice and Experience 29 (8), 721–740.

Malmi, L., Karavirta, V., Korhonen, A., Nikander, J., Seppl, O., Silvasti, P., 2004. Visual algorithm simulation exercise system with automatic assessment: TRAKLA2. Informatics in Education 3 (2), 267 – 288.

Saikkonen, R., Malmi, L., Korhonen, A., 2001. Fully automatic assessment of programming exercises. In: Proceedings of The 6th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education, ITiCSE'01. ACM, Canterbury, United Kingdom, pp. 133–136.

Vihtonen, E., Ageenko, E., 2002. Viope-computer supported environment for learning programming languages. In: The Proceedings of Int. Symposium on Technologies of Information and Communication in Education for Engineering and Industry (TICE2002). Lyon, France, pp. 371–372.

A TRAKLA2 Exercises

Table 3: The visual algorithm simulation exercises in TRAKLA2 system. The column *name* describes the topic and the *description* characterizes the exercise. The roman numbers (i-iv) indicate the separate exercises and the number of sub-topics.

Name	Description
Insertion into (i) Binary search tree, (ii) Digital search tree, and (iii) Radix search trie	The learner is to insert random keys into an initially empty search tree by dragging and dropping them into the correct positions.
Binary search tree deletion	The learner is to remove 4 keys from a binary search tree of 15 to 20 keys. Pointer operations are simulated by directly manipulating the edges that connect the nodes of the tree in the visualization.
Faulty Binary Search Tree	The learner is to show how to bring the following binary search tree in an inconsistent state: duplicates are allowed and inserted into the left branch of an equal key, but the deletion of a non-leaf node relabels the node as its successor.
AVL tree (i) insertion, (ii) single rotation, and (iii) double rotation	The learner is to (i) insert 13 random keys into an initially empty AVL-tree. The tree (i-iii) has to be balanced by rotations. The rotation exercises (ii-iii) require pointer manipulation, while the insertion exercise (i) provides push buttons to perform the proper rotation at the selected node.
Red-black-tree insertion	The learner is to insert 10 random keys into an initially empty Red-Black-tree. The color of the nodes must be updated and the tree must be balanced by rotations.
BuildHeap algorithm	The learner is to simulate the linear time buildheap algorithm on 15 random keys.
Binary heap insertion and delete min	The learner is to a) insert 15 random keys into a binary heap and b) perform three deleteMin operations while preserving the heap order property (see Fig. 1).
Sequential search: (i) Binary search, and (ii) Interpolation search	The task is to show which indices the algorithm visits in the given array of 30 keys by indicating the corresponding keys.
Tree traversal algorithms: (i) preorder, (ii) inorder, (iii) postorder, and (iv) level order	The learner is to show which keys in a tree the algorithm visits by indicating the visited keys in the required order.
Preorder tree traversal with stack	The learner is to simulate how the stack grows and shrinks during the execution of the preorder algorithm on a given binary tree.
Fundamental Graph algorithms: (i) Depth First Search, and (ii) Breadth First Search	The learner is to visit the nodes in the given graph in DFS, and BFS order.
Minimum spanning tree algorithms: Prim's algorithm	The learner is to add the edges into the minimum spanning tree in the order that Prim's algorithm would do.
Shortest path algorithms: Dijkstra's algorithm	The learner is to add the edged to the shortest paths tree in the order that Dijkstra's algorithm would do.
Open addressing methods for hash tables: (i) linear probing, (ii) quadratic probing, and (iii) double hashing	The learner is to hash a set of keys (10-17) into the hash table of size 19.
Sorting algorithms: (i) Quicksort, and (ii) Radix Exchange sort	The learner is to sort the target array using the given algorithm.