CONTENT DOMAIN AND STRATEGY FOR TEACHING LINKED LIST IN PEDAGOGICALLY EFFECTIVE MANNER

M.S. Farooq, A. Abid, U. Farooq and I. Saleem

Department of Computer Science, University of Management and Technology, Lahore, Pakistan
Corresponding author's email: shoaib.farooq@umt.edu.pk

ABSTRACT: The objective of this study was to address the problem of effectively teaching linked list, which was a core topic of data structure course. For this purpose, a survey based methodology was adopted to define the content domain and a teaching strategy for this topic. Among the contributions of this research, firstly, the content domain of the topic was defined using a taxonomy comprising of sub categories that included structure of a linked list; implementation variants; persistence of a linked list; and advanced complex variants. Secondly, the relative importance of the sub-topics in the content domain was also defined by conducting a survey form teachers and students. It was observed that out of thirteen subtopics, only seven should be covered in the first course of data structure, while the rest of the six topics could be provided as optional reading material.

Keywords: Abstract data type, Sentinel linked list, Variants of link list, Object oriented and structured paradigms.

(Received 06-04-2016 Accepted 15-01-2018)

INTRODUCTION

Linked list is a core data structure and an essential part of a computer programming course being taught in the curricula of computer science and electrical engineering disciplines (Cassel et al., 2008). A linked list is defined as a collection of nodes that can be traversed starting from the first node, generally known as head of the list. Each node of a linked list comprises of a data part and an address part, where the address part points to the next node in the list (Blevins, 2009). Linked list is very useful in situations where the program needs to manage memory dynamically based on non-contiguous blocks (Schwartz, 1990). A linked list is considered more flexible than arrays as it can grow and shrink on demand dynamically. New nodes can be inserted between any two nodes seamlessly without disturbing or moving any other data elements (Yunhong, 2015).

The concept of linked list was introduced by Allen Newell, Cliff Shaw and Herbert in 1955 as core data structure for Information Processing Language (IPL).

(McCarthy, 1978). The LISP (List Processor) was developed in 1958 at MIT by John McCarthy (McCarthy, 1960). ACM curriculum 2001, 2008 and 2013 proposes linked list as major topic in courses such as fundamental of data structures, data structures and algorithms, parallel algorithms and system programming (Roberts *et al.*, 1999).

Linked list holds a central importance in the course of data structures. Almost all text books on data structure cover linked list topic as a core part of data structure and algorithms course. From implementation viewpoint, most of the text books follow either structured or object oriented programming paradigm to teach all the data structure (Drozdek, 2012, Dastidar *et al.*, 2003 and Goodrich *et al.*, 2007). Most of the famous text books contain relatively difficult codes listings to understand the implementation details (Malik, 2016). In a study carried out by (Dastidar *et al.*, 2003) involving the use of friend functions for implementation of linked list.

The Linked list is classified according to different compositions of its nodes (Dastidar *et al.*, 2003). One such variant is called singly linked list in which each node comprises of a data part and a pointer to the next node. Whereas, the other variant is called doubly linked list where each node comprises of a data part, but includes two pointers, one pointing to the previous node in the list and other pointing to next node.

In terms of its implementation linked list can be categorized into imperative, circular, abstract data type (ADT), generic, sentinel and array based linked lists. All these variants offer different useful ways in which a linked list is implemented (Sahami *et al.*, 2012 and Malik, 2016).The storage dimension involves temporal persistence of a linked list. The data of a linked list is stored on the hard disk to make it persistent. Whereas, on the other hand, data remains in the main memory for a certain time based on the scope of the object, called volatile list (Drozdek, 2016).The concept of generic linked list was introduced by Blevins in FORTRAN 95 which can store arbitrary data types (Blevins, 2009).

The ordered Linked list was best for search, update, and delete operations. Pual F. Dietz introduced two major algorithms for ordering i) ordering through ancestor relationship, ii) ordering through maintains a tree structure environment. Both have O (nlgn) complexity for search, update and delete operations (Dietz, 1982). The concept of skip list was introduced as ordered probabilistic linked list data structure in 1989 as alternative to balanced and binary search tree (Pugh, 1990).

Major aim of this research was to design an effective strategy for the learning of linked list in a best pedagogical manner. There was a need to select which core topics of linked list should be taught. There were so many variants of Linked list as shown in Figure-1; it was very hard for a teacher to choose best variant, sequence and pedagogy of leaning related to linked list taxonomy. A survey was conducted from two stakeholders' teachers and students on linked list taxonomy. Teachers were asked to select importance level and teaching level difficulty of linked list. Students were to select importance level and level of learning difficulty.

MATERIALS AND METHODS

This research employed a comprehensive literature study of data structures text books and relevant articles. These artifacts worked as the main source of material for the proposed study. All these topics and subtopics related to linked list were then processed to define taxonomy of all different variants of linked list. This taxonomy also served as the first contribution of this research work.

Two separate questionnaires were developed for teachers and students. The questionnaire for teacher consisted of two parts; first part contained some questions in order to get the broad level viewpoint of the teacher. Whereas, in the second part of the questionnaire teachers were asked to share their views about each topic and subtopic of linked list regarding its level of importance, and the level of difficulty in terms of teaching.

Similarly, another questionnaire was designed for the students. The questionnaire for students contained some questions regarding importance and learning difficulty for subtopics. In this study 25 teachers and 96 students were involved. Data obtained from the teachers was processed as shown in Table 1, and from the students as shown in Table 2.

Criterion for topic selection using weighted sum was shown in Table 3. This resulted into selection of certain topic selection, as well as, removal of certain topic and subtopic. This selection and removal process resulted into a smaller subset of the linked list topic. This subset was considered to be easy to teach from a teacher's perspective, and easy to learn from a student's perspective.

The scores given by the teachers to a given subtopic 'c' are aggregated in the variable C^T, while the score given by the students to a given subtopic 'c' are aggregated in the variable C^s. The weighted scores range from [1, 3]. Thus the score is divided by 3 to keep the score normalized. The following equation is used to compute aggregated score for a given subtopic 'c'.

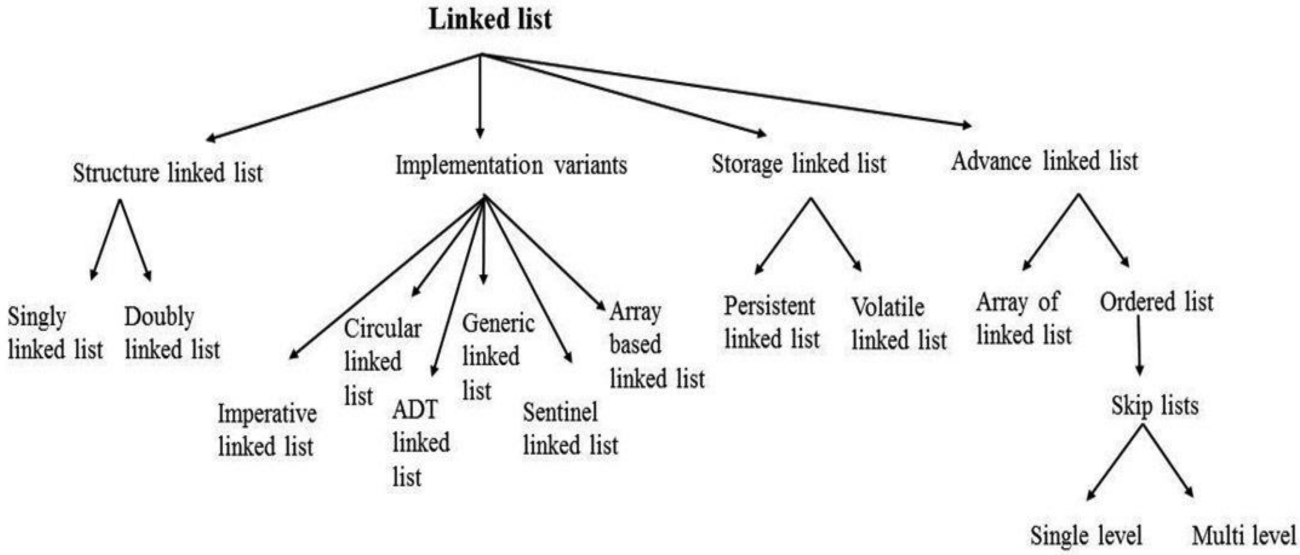
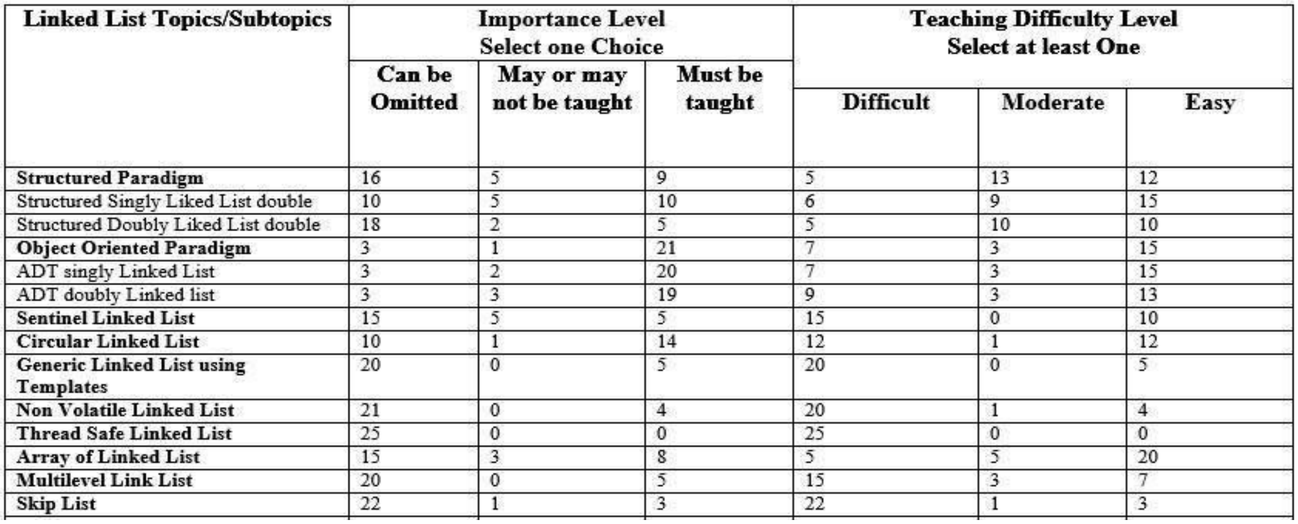


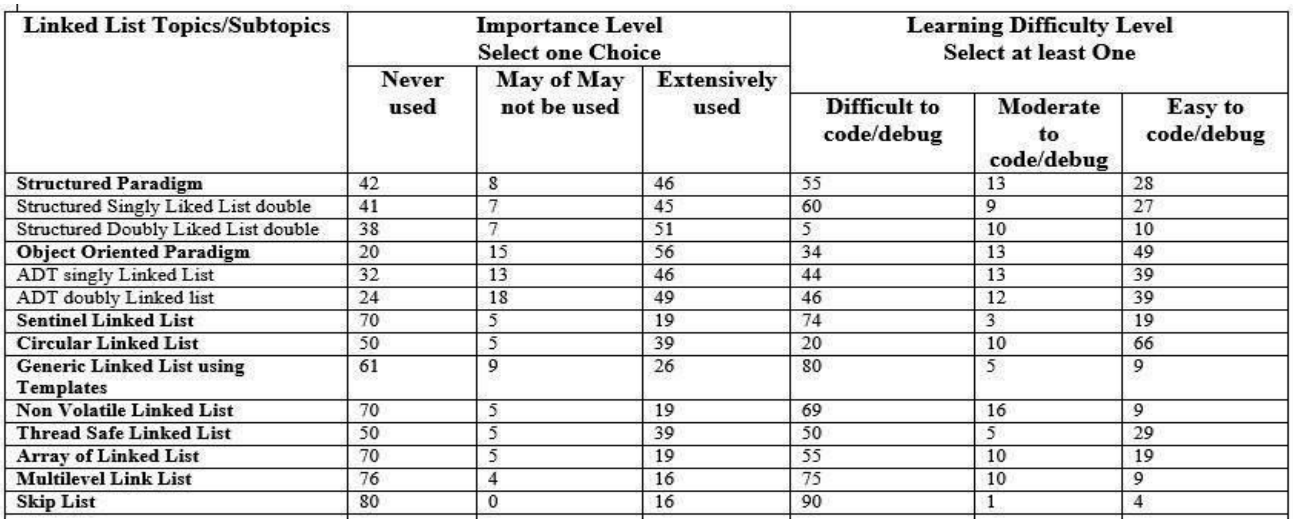
Figure-1: Linked list taxonomy

Table-1: Data collected from teachers about the importance and teaching difficulty for subtopics



Linked List Topics/Subtopics	Importance Level Select one Choice			Teaching Difficulty Level Select at least One		
	Can be Omitted	May or may not be taught	Must be taught	Difficult	Moderate	Easy
Structured Paradigm	16	5	9	5	13	12
Structured Singly Liked List double	10	5	10	6	9	15
Structured Doubly Liked List double	18	2	5	5	10	10
Object Oriented Paradigm	3	1	21	7	3	15
ADT singly Linked List	3	2	20	7	3	15
ADT doubly Linked list	3	3	19	9	3	13
Sentinel Linked List	15	5	5	15	0	10
Circular Linked List	10	1	14	12	1	12
Generic Linked List using Templates	20	0	5	20	0	5
Non Volatile Linked List	21	0	4	20	1	4
Thread Safe Linked List	25	0	0	25	0	0
Array of Linked List	15	3	8	5	5	20
Multilevel Link List	20	0	5	15	3	7
Skip List	22	1	3	22	1	3

Table-2: Data collected from student about the importance and learning difficulty for subtopics.



Linked List Topics/Subtopics	Importance Level Select one Choice			Learning Difficulty Level Select at least One		
	Never used	May of May not be used	Extensively used	Difficult to code/debug	Moderate to code/debug	Easy to code/debug
Structured Paradigm	42	8	46	55	13	28
Structured Singly Liked List double	41	7	45	60	9	27
Structured Doubly Liked List double	38	7	51	5	10	10
Object Oriented Paradigm	20	15	56	34	13	49
ADT singly Linked List	32	13	46	44	13	39
ADT doubly Linked list	24	18	49	46	12	39
Sentinel Linked List	70	5	19	74	3	19
Circular Linked List	50	5	39	20	10	66
Generic Linked List using Templates	61	9	26	80	5	9
Non Volatile Linked List	70	5	19	69	16	9
Thread Safe Linked List	50	5	39	50	5	29
Array of Linked List	70	5	19	55	10	19
Multilevel Link List	76	4	16	75	10	9
Skip List	80	0	16	90	1	4

Table-3: Criterion for topic selection using weighted sum

Weightage		**1**			**2**		
τ	**Teachers (T)**	**Low**	**Medium**	**High**	**Low**	**Medium**	**High**
	Choose One	**1**	**2**	**3**	**1**	**2**	**3**
σ		**Conceptual Importance (CI)**			**Learning Difficulty (LD)**		
	Students (S)	**Low**	**Medium**	**High**	**Low**	**Medium**	**High**
	Choose One	**1**	**2**	**3**	**1**	**2**	**3**

$_{C}$ ($_{C}^{T}$/ 3) ($_{C}^{S}$ / 3)

Where, τ is the overall weightage given to the opinion of a teacher, and σ is the weightage given to the opinion of a student.

RESULTS AND DISCUSSION

Table-4 presented the selected and discarded topics and subtopics of the linked list given in taxonomy Figure-1. Here topics and subtopics were presented in the form of constructs in different groups according defined taxonomy.

Based on the survey results, topics discussed in this taxonomy have been divided into three different categories based on their importance for the course. These categories included core topics, supportive topics, and optional topics. Where, core topics were the ones which must be taught by the instructors in the class in detail. Supportive topics were the ones which should be covered in the respective lab sessions or assignments given to the students. Lastly, the optional topics were the ones which could be provided to the students in terms of relevant reading material, or could be taught in the advanced courses of data structure.

Many different ways of teaching using scaffolding have been introduced in the past. For instance, the iList was an Intelligent Tutoring System (ITCs) tool which helps novices to learn linked list within an interactive and user friendly environment (Fossati et al., 2008 and Fossati *et al.*, 2009). Huggins introduced Barrel of Monkeys toy to teach the concept of Linked Listing at Ketttering University (Huggins, 2005). David Furcy developed JHAVEPOP, a visualization tool for elementary pointer and linked list operations (Furcy, 2009). Herbert developed JVALL, a software package that provides an animation of all core linked list operations (Dershem *et al.*, 2002).However, this research categorized the subtopics into different levels of importance for teaching the topic of linked list.Table-5 presented the details of contents that were discussed in this article. The first column presented the respective head of details, based on the already defined taxonomy. In terms of structure it was suggested that both singly and doubly linked lists should be covered as core topic. Whereas, in terms of implementation, two variants of teaching this course have been discussed. Firstly, it can be taught using imperative approach which was based on structure programming and involved imperative first and object oriented approach later. In this case, the topics were covered without involving the concepts of object oriented paradigm. Therefore, the topics of singly and doubly linked list using structures were considered as core topics; circular linked list was considered in the category of the supporting topics; whereas, Sentinel and Array based linked list was considered in the section of optional topics using this programming paradigm. Secondly, in terms of object oriented paradigm, the concept of ADT was

Table-4: Selected and discarded topics/subtopics

Linked List Topics	Selected/ Subtopics	Score	Rejected/Subtopics
Structured Singly Linked List	Yes (addNode, removeNode, insertNode)	0.90	
Structured Doubly Linked List	Yes (addNode, removeNode, insertNode)	0.81	
Circular Linked List	Yes (addNode, removeNode, insertNode)	0.81	
Object Oriented Linked List	Yes (addNode, removeNode, insertNode, Traverse, addOnHead, addOnTail)	0.84	
Generic Linked List		0.54	Rejected
Sentinel Linked List	Yes(addNode, removeNode, insertNode, Traverse, addOnHead, addOnTail)	0.72	
Array Based Linked List		0.45	Rejected
Persistent Linked List		0.54	Rejected
Volatile Linked List		0.51	Rejected
Array of Linked List	Yes (addNode, removeNode, Traverse)	0.69	
Single Level Skip List		0.33	Rejected
Multilevel Level Skip List		0.21	Rejected

involved while teaching the core topics of singly and doubly linked lists. Whereas, generic linked list was considered in the supporting topics; and lastly, sentinel linked list and array based implementation of linked list using object oriented paradigm which was considered in among the optional topics using this paradigm as shown in table-5.

Table-5: Suggested importance of the topics under each category.

Topic		Core Topics	Supporting Topics	Optional Topics
Structure		**Singly linked list, doubly linked list**		
Impleme-ntation	**Structured Paradigm**	**Imperative linked list (both Singly and Doubly)**	**circular linked list**	**Sentinel linked list, Array based**
	Object Oriented Paradigm	**ADT linked list (both Singly and Doubly)**	**Generic linked list**	**Sentinel linked list, Array based**
Storage linked list		**Volatile linked list**	**Persistent linked list**	
Complex/advance Variants of linked list			**Array of linked list, Ordered/ Skip list**	**Multi-level Skip lists**

The time allocation for teaching these topics to the students was left at the disposal of the instructor, as this issue involved several different factors i.e. the quality of students, grasp of the instructor on the course etc.

Conclusion: A survey based methodology was adopted to define a subset of topics that should be taught in an introductory course of data structure. It was observed that out of thirteen subtopics, only seven should be covered in the first course of data structure, while the rest of the six topics could be provided as optional reading material.

REFERENCES

Blevins, J.R. (2009). A generic linked list implementation in FORTRAN 95. Assoc. for Com. Mach. Fort. For. 28(3): 2-7.

Cassel, L., Clements, A., Davies, G., Guzdial, M., McCauley, R., McGettrick, A., and Weide, B.W. (2008). Computer science curriculum 2008: An interim revision of CS 2001. Assoc. for Com. Mach. 31(8): 1992.

Dastidar, G.D., Chattopadhyay, M., Chattopadhyay, S., and Ghosh, D. D. (2003). Data Structures Through C Language. 1st Ed. BPB publication;Delhi (India). 69 p.

Dershem, H.L., McFall, R. L., &Uti, N. (2002). Animation of Java linked lists. Assoc. for Com. Mach. 34(1): 53-57.

Dietz, P.F.(1982). Maintaining order in a linked list. In Proceedings of the fourteenth annual symposium on Theory of computing Assoc. for Com. Mach. 122-127.

Drozdek, A. (2016). Data Structures and algorithms in C++. 4th Ed. Cengage Learning; Boston(USA). 98 p.

Fossati, D., Di Eugenio, B., Brown, Ohlsson, Cosejo, (2009). Supporting computer science curriculum: Exploring and learning linked lists with iList. IEEE Trans. on Learn. Tech. 2(2): 107-120.

Fossati, D., Di Eugenio, B., Brown, C., &Ohlsson, S. (2008). Learning linked lists: Experiments with the iList system. International Conference on Intelligent tutoring systems, Berlin(Germany). 5091: 80-89

Furcy, D. (2009). JHAVEPOP: visualizing linked-list operations in C++ and Java. Jr. of Comp. Sci. in Coll. 25(1): 32-41.

Goodrich, M., Tamassia, R., and Mount, D.(2015).Data Structures And Algorithms In C++. 2nd Ed. John Wiley & Sons, New Jersey(USA). 136-137 p.

Huggins, J.K. (2005). Using a Barrel of Monkeys in computer science. Proccedings Frontiers in Education 35th Annual Conference. Indiana(USA). F4C-1.

H., Yunhong, L., Jinglei, Z., Xiaoming, T., & Liu, C. (2015) Application of Dynamic Linked List in the Software Design of Power Test System Microprocessors.2015(2).

Malik, D.S. (2016). C++ programming: Program design including data structures. Cengage Learning; Boston(USA). 104-107 p.

McCarthy, J. (1960). Recursive functions of symbolic expressions and their computation by machine, Part I. 3(4): 184-195.

McCarthy, J. (1978). History of LISP. In History of programming languages I. Assoc. for Com. Mach. Newyork(USA). 173-185.

Pugh, W.(1990). Skip lists: a probabilistic alternative to balanced trees. Communications of the ACM. 33(6): 668-676.

Roberts, E., Shackelford, R., LeBlanc, R., & Denning, P. J.(1999). Curriculum 2001: Interim report from the ACM/IEEE-CS task force Assoc. for Com. Mach. SIGCSE Bulletin. 31(1):343-344

Sahami, M., Roach, S., Cuadros-Vargas, E., & Reed, D. (2012). Computer science curriculum 2013: reviewing the strawman report from the ACM/IEEE-CS task force. In Proceedings of the 43rd technical symposium on Computer Science Education. Assoc. for Com. Mach. North Carolina(USA). 3-4 p.

Schwartz, A. (1999) "Method and system for accessing an item in a linked list using an auxiliary array." U.S. Patent 5,950,191, issued September 7, 1999.