Operation of Queue and Stack by DNA Tiles

Xinxin Zhang[1,a*], Nan Zhao[2,b], Jing Yang[3,c]

[1]School of Control and Computer Engineering, North China Electric Power University, Beijing China
[2]School of Control and Computer Engineering, North China Electric Power University, Beijing China
[3]School of Control and Computer Engineering, North China Electric Power University, Beijing China

Abstract. DNA is used as self-nanomaterials to assemble into specific structures. DNA tile provides a new idea for the application of DNA tile in the field of computing. Recent years, Queue and Stack are important linear data structures which are used in various software systems widely. The implementation of DNA based queue and stack has been studied continuously for many years. In the traditional DNA computing, queue and stack are mostly realized by DNA strands displacement, restriction endonuclease and ligase were used. However, as an active material, it has a high requirement for enzyme experimental conditions. The purpose of this paper is to implement queue and stack structures using non-enzyme systems. The rule of Queue is characterized by FIFO (first in first out), which allows for insertion at one end of the list and deletion at the other. The rule of Stack is characterized by FILO(first in last out), which allows for insertion and deletion at one end of the list. We are aimed to implement Queue and Stack using self-assembly and disassembly via DNA Tiles. No enzymes are needed for the whole experiment. As an enzyme-free system, it provides a new method to implement stack and queue.

1 Introduction

A DNA computer is a complete form of information technology that uses DNA (deoxyribonucleic acid) to solve complex mathematical problems through molecular biological operations using coded DNA sequences (usually computer memory) as objects. Computer programs are inseparable from a data structures. Queue and Stack are important linear structures which are used in various software systems widely. The research on the implementation and application of Queue and Stack has never stopped. Here, we use DNA tiles to implement Computer data structure (queues and stacks) by a completely new way, as shown in figure 1.

Throughout the research, most of the Queue and Stack are implemented by strand replacement, estriction endonuclease and ligase, etc. However,enzyme as a kind of active material, has high requirements for experimental conditions. Enzyme-free systems have not been studied. In the 1990s, Seeman et al. designed a variety of DNA cross-structure known as DNA Tiles1. In 1998, Erick Winfree proposed DNA blocks (DNA Tile) as "Wang" Tile to store data2. It provides a new idea for the application of DNA Tile in computing field. This paper is aimed to implement Queue and Stack using self-assembly and disassembly via DNA Tiles.

DNA Tile is the basic unit of DNA assembly3. It has a variety of unit structures4. DNA Tiles have solved a lot of problems since they were introduced9. However,there has been little involvement in implementing storage structures within data structures using DNA Tiles. In order to solve the design and implementation of the Queue and Stack data structures, the self-assembly and disassembly method of DNA proposed by Hari k. k. and Elisa Franco13. The proposed method is improved and innovated to complete the design of the Queue and Stack.
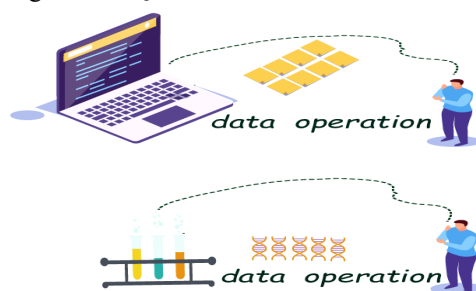

Fig1. Schema of using DNA tile to implement computer data structure.

2 DNA tile

We consider DNA double-crossover tiles formed by four unique DNA strands, as shown in figure 2a. Tiles assemble into linear structure via programmed interactions of their sticky ends (domains marked a,b,a' and b'). The function of strand d is to store data as a storage unit. Tiles were modified to include two single-stranded overhang, or toehold, illustrated as the black domain on the 5' end of the yellow strand. Hybridization of sticky ends results in the formation of linear, toeholds placed on the 5' end of the yellow strand are expected to be on the external surface of the liner. The presence of a toehold

---

[a]yjzcdd_2000@ncepu.edu.cn, 547419699@qq.com, [b]974500359@qq.com, [c]<yjzcdd_2000@ncepu.edu.cn>

makes the neighbouring sticky end accessible for strand invasion: an invader molecule complementary to the toehold and the sticky end domain invades one of the inter-tile bonds, causing the linear to disassemble and making tiles inactive. If it includes an additional toehold domain, the invader strand can be displaced by an anti-invader strand, which reactivates tiles, enabling their reassembly into linear, like figure 2b shown.
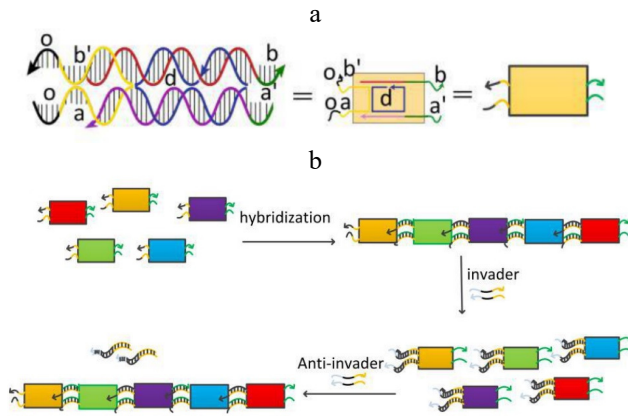

Fig2. The assembly and disassemby of DNA Tile. (a)Details of DNA tiles,DNA tiles formed by four unique DNA strands. (b)Tiles assemble into linear structure via the sticky a, b, a'and b'.Domain black are expected to be on the external surface of the liner. An invader invades inter-tile bonds, causing the linear to disassemble and making tiles inactive.the invader strand can be displaced by an anti-invader strand, which reactivates tiles, enabling their reassembly into linear.

3 Queues and tacks

Queues and tacks are common data structures. It plays an important role in all fields of computing. It is the basis of algorithm design and calculation, and it is also indispensable for the computer of new molecular computing mode. At home and abroad in the field of study, in 2007, Queue data structure in the computer design method is proposed by Wanggen Li, Yongsheng Ding, this method is to use the double strand molecules as a Queue in the computer storage structure14, with two primers called Primer_L and Primer_R designed for PCR reaction. Two different restriction enzymes was used to complete operation of Enqueue and Dequeue . Yali Zhu, Ken Li et al. proposed that under the action of ligase, nodes will form DNA double strand by hybridization and ligation reactions15.All the biotechnology used in them can be realized in the laboratory. These methods lay a solid foundation for the combination of computer science and biological science. However, in these methods, various restriction enzymes and ligases are basically used, making experiment has certain limitations. To sum up, it is necessary to further study the storage and representation of data structure as well as its operation for the practical application of computers.

The rule of Queue is characterized by FIFO(first in first out), which allows for insertion at one end of the list and deletion at the other. The rule of Stack is characterized by FILO(first in last out), which allows for insertion and deletion at one end of the list, figure 3. They're all common linear structures.
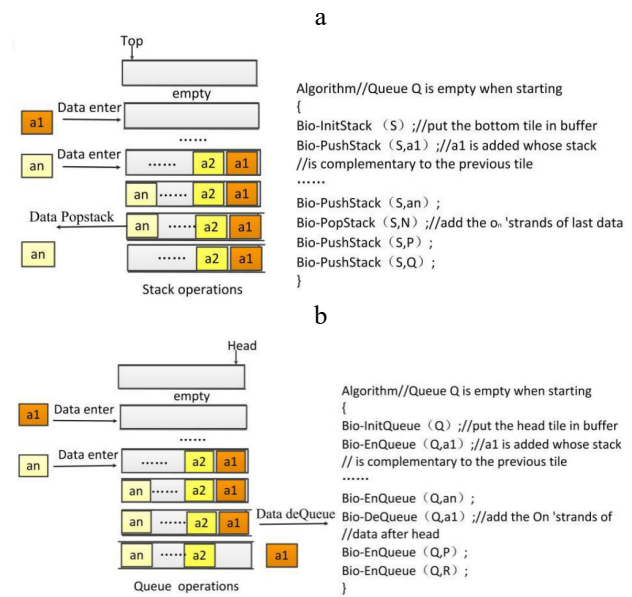

Fig3. Operation of Queue and Stack. (a)A Queue is a special type of linear list that allows for insertion at one end of the list and deletion at the other. The end of Queue allowed delete operation is called the Queue Head. (b)Stack is a linear list that restricts inserts and deletes data only on one side. The end that allows inserts and deletions is called the top of the Stack.

4 Operations of Queue and Stack

The DNA tile acts as a data storage unit, we made the following stipulations: domain "on" is fixed length and variable sequence, where different "on" represents different tiles (data) to implement the operation of a particular tile (data). The function of the part containing N is to store data as a storage unit . The length and sequence of N can be changed according to the amount of we want. Like figure 4a shown, Queue is a special type of linear list that allows for insertion at one end of the list and deletion at the other. The end of the allowed insert operation is called Queue Rear, and the end of the allowed delete operation is called the Queue Head. This structure is characterized by FIFO. The Head and Rear of the Queue are indicated by the Head and Rear Pointers respectively in data structure. When there is no data in the Queue, it is called an empty Queue. Because DNA has massive storage capacity , overflow of Queues is not considered.

To initialize the Queue in the data structure, a Queue Head pointer will be set. When the data is out of the Queue, the Queue Head pointer will point to the next data. Inspired by this condition, Head-DNA Tile is created, as shown in figure 4(b). The Head-DNA Tile consists of four single strands of DNA, which are part of the DNA Tile, with only the a, b' sticky end remaining to stick the first data. All sequences are fixed in Head-DNA Tile16. If only Head-DNA Tiles without DNA tile in the entire system indicate that the Queue is empty.
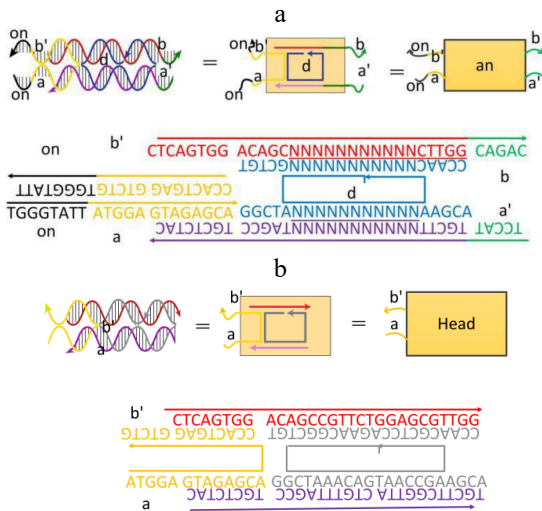
Fig4. (a)The function of the part containing N is to store data as a storage unit . The length and sequence of N can be changed according to the amount of we want. (b)DNA Head tile used in this study.Structural schematic showing the intertwining of the DNA Head strands. Five unpaired bases at leftmost (yellow) strands are sticky ends, responsible for directing tile assembly. We simplify tiles to rectangles. Unlike DNA Tile, there is no storage domain in head tile.

4.1 Enqueue

Queue entry refers to data entering one after another to form a sequence structure, as shown in figure 5a, indicating the process of Queue entry. DNA Tiles are macromolecules made of multiple intertwined strands of synthetic oligonucleotides, Whose sequences are designed to place The binding sites in definite positions and orientations relative to one another using immobile branched junctions8. The predictability and specificity of Waton-crick base pairing thus enabling self-assembly of extended structures with nanometer resolution and tunable stability. Five unpaired bases at both ends of the rightmost (green) and leftmost (yellow) strands are sticky ends, responsible for directing Tile assembly13. Figure5b show Enqueue by DNA Tiles.
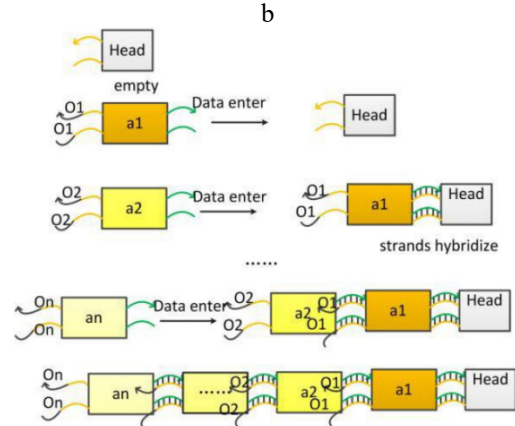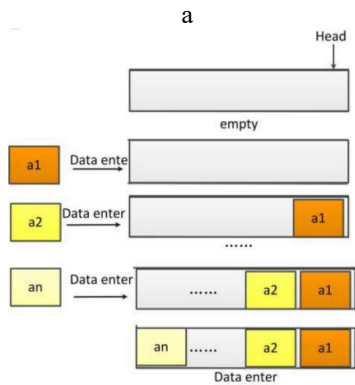

Fig5. (a)tructural schematic showing the Enqueue in data structure (b) tructural schematic showing the Enqueue by DNA Tile:five unpaired bases at both ends of the rightmost (green) and leftmost (yellow)Strands are sticky ends, responsible for directing Tile assembly.

4.2 Dequeue

Dequeue refers to data detach from the Queue in sequence, as shown in Figure6a. the different sequence of domain o is used to indicate specific data. As shown in the figure 6b, when we want to get one data, we need to add the invader strand to the DNA tile whose connected with DNA-Head Tile. When the invader is not consumed, the Queue is empty.After the Dequeue, the Head-DNA Tile is added again to indicate the first data of the Queue.
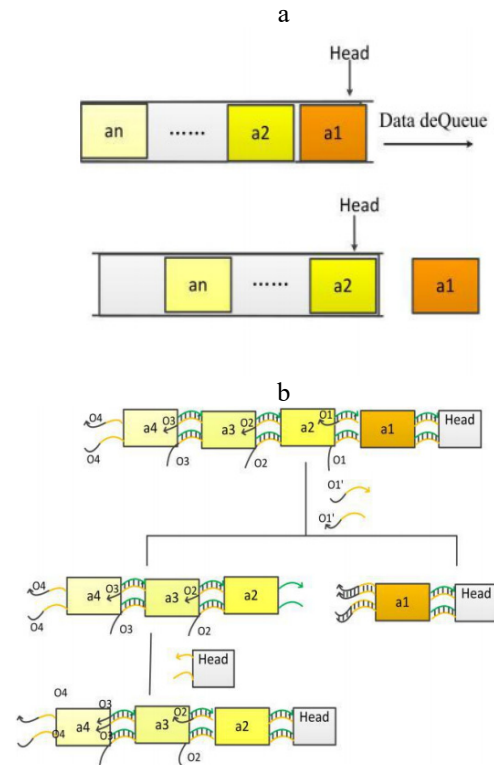

Fig6. (a) tructural schematic showing the Dequeue in data structure (b) add the invader strand to the DNA tile whose connected with DNA-Head Tile. When the invader is not consumed, the Queue is empty. Once data Enqueue, the Head-DNA Tile is added again to indicate the first data of the Queue.

4.3 Definition of Stack

A Stack is a linear list that restricts inserts and deletes data only on one side. The end that allows inserts and deletions is called the top of the Stack, and the end that does not allow inserts and deletions is called the bottom of the Stack. An empty list without data is called an empty Stack. The main difference between the two construction is that the Stack is first in last out, while the Queue is first in first out. Similar to Queue initialization, we still use the Head-DNA Tile structure, but at the Stack, we call it the Bottom-DNA Tile. Only the Bottom-DNA Tile in the system indicates that the Queue is empty. Queues are FIFO principles, Stacks are FILO principles, there is no difference when they enter the structure, so we won't talk about Pushstack.

4.4 Popstack

Popstack is an operation in which data leaves the Stack according to certain rules, as shown in figure 7a. Since the rule of the Stack is FILO, we need to add the invader of the last data on the top of the Stack when Popstack. Since the operation at the top of the Stack , there is no need to add a Bottom-DNA Tile, as shown in figure 7b.
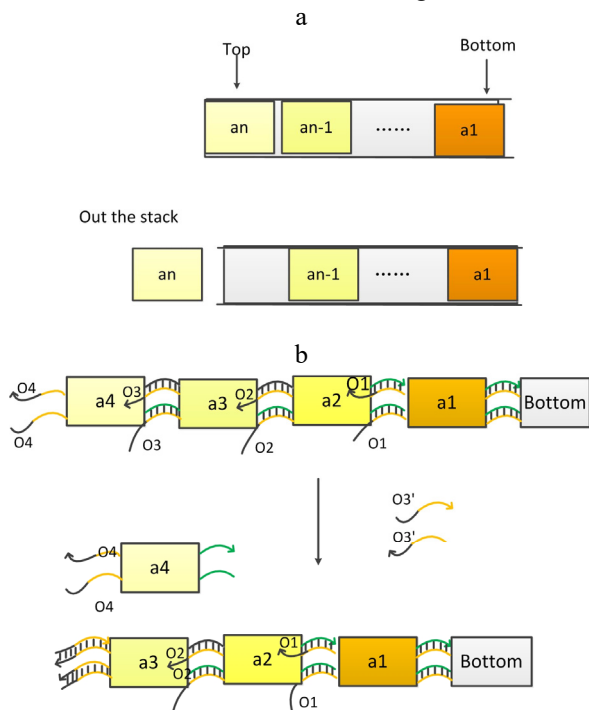

Fig7. (a) tructural schematic showing the Popstack in data structure (b) tructural schematic showing the Popstack by DNA Tile. The rule of the Stack is FILO, so we need to add the invader of the last data on the top of the Stack when PopStack. Since the operation at the top of the Stack, there is no need to add a Bottom-DNA Tile.

5 Design of arrays by DNA tiles

We were inspired by the anti-invader[13]proposed by Subramanian and Elisa Franco have designed operations of array. An array is a finite sequence of data which is the same type, as shown in figure 8 is an array including six

data. The order of data elements in an array is fixed. Once an array is defined, the number of data elements is no longer changed. The data structure in the array has the same data type . Each data in the array corresponds to a unique coordinates. An array is a storage structure that allows random access to any data in an array through a[]. Figure 8 is read a1 randomly. When access a data at random In array, we can extract this data by adding invader of its and the next. Invader is complementary to the domain o, and matches the address of the array "an", and here introduce the concept of anti-invader. The mechanism of DNA tile invasion has been designed to be reversible: by including a toehold "t" on the invader strand, it is possible to displace it from tiles using a complementary anti-invader strand including "t". By displacing and titrating the invader, the anti-invader restores the ability of tiles to nucleate and polymerize13. Figure 8b is read a1 by DNA tile, using invader of a1 and a2 to get a1 for DNA-seq, and using anti-invader make a1 return to the array.
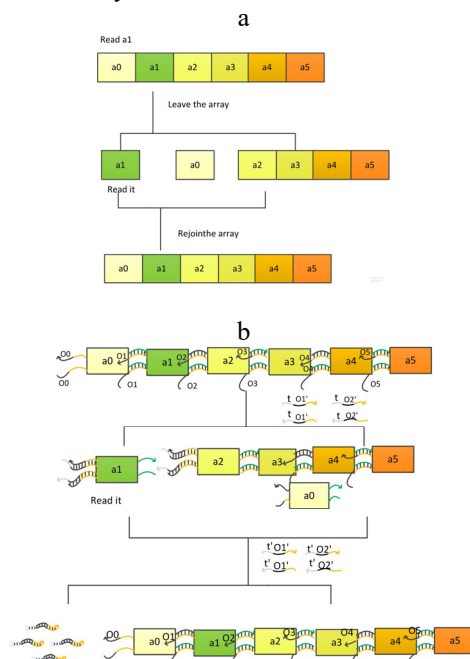

Fig8. (a)array including six data, read a1.(b) read a1 by DNA tile, using invader of a1 and a2 to get a1 for DNA-seq, and using anti-invader make a1 return to the array.

6 Conclusion

The basic principles of the data structure in the computer of DNA computing of research at home and abroad are understood and mastered at first. It is concluded that current techniques for queue and stack implementation using DNA computing all require restriction endonuclease as well as ligase, polymerase, etc. As an active organism, enzymes have higher requirements on the living environment. Considering the stability, robustness and operability of the DNA tile, we modified the DNA tile mentioned by Franco, and changed it from five DNA strands to four, so as to realize the goal of forming a linear structure from self-assemby. DNA tiles use their sticky ends to realize self-assembly and realize data entry in linear structures. By adding invader strands to realize the

disassembly of DNA tile, the output of data in the data structure is realized. The difference from the previously work, where different On of DNA strands represents different tiles (means datas in a data structure) to implement the operation of a particular tile (data). Winfree proposed DNA tiles as "Wang" tiles to store data, marking more applications for DNA tiles in computing. In this paper, the queue and stack and are realized by DNA tiles. As an enzyme-free system, it provides a new method to implement queue and stack.

ACKNOWLEDGMENTS

REFERENCES

1. Adleman L M(1994).Molecular computation of solutions to combinatorial problems.Science,266 (5187),1021-1024.

2. Winfree E(1998).Algorithmic self-assembly of DNA [Degree]. CaliforniaCA: California Institute of Technology.

3. Haiyan Z.,Xiaolong S(2019).DNA Tile calculation . Journal of guangzhou university18 (2),65-74.

4. Fu T J.Seeman N C(1993).DNA double-crossover molecules . Biochemistry32 (13),3211-3220.

5. LaBean T H.Yan H.Kopatsch J.et al(2000). Constructionanalysisligationand self-assembly of DNA triple crossover com-plexes . Journal of the American Chemical Society122 (9),1848-1860.

6. Yan H.Park S H.,Finkelstein G.,et al(2003). DNA - templated self - assembly of protein arrays and highly conductive nanowires.Science,301 (5641),1882-1884.

7. Yin P.,Hariadi R F.,Sahu S.,et(2008). alProgramming DNA tube circumferences . Science,321 (5890),824-826.

8. Liu W.,Zhong H.,Wang R.,et al(2011).Crystalline two-dimensional dna-origami arrays .Angewandte Chemie International Edition,50 (1),264-267.

9. Winfree E.Liu F.,Wenzler L A(1998).Design and self-assembly of two-dimensional DNA crystals.Nature,394 (6693),539.

10. Sherman W B.,Seeman N C(2004).A precisely controlled DNA biped walking device.Nano Letters,4 (7),1203-1207.

11. Aldaye F A.,Sleiman H F(2007).3 d discrete 'DNA assemblies Modular access to structurally switchable.Journal of the American Chemical Society,552 (7683),13376-13377.

12. Tikhomirov G.,Petersen P.,Qian L(2017).Fractal assembly of microshoe-scale DNA origami arrays with spatial patterns.Nature,552 (7683),67.

13. Hari k.k,Elisa Franco(2019).The Autonomous dynamic control of DNA nanostructure self - assembly. NatureChemistry,11,510-520.

14. Wanggen L.,Yongsheng D(2007).Design and implementation of Queue data structure in DNA computer .Journal of computer science,30 (6), 993-998.

15. Zhu.,et al(2008).strand storage structure of binary tree in DNA computer . Computer application research,25 (09), 2632-2636.

16. Fut.Seeman(1993).n. DNA double-crossover molecules. Biochemistry,32,3211-3220.

17. Rothemund. et al(2004).The Design and characterization of nanotubes programmable DNA . Am. Chem. Soc,12616344-16352.