

<https://doi.org/10.7236/JIIBC.2024.24.1.149>
JIIBC 2024-1-22

단일 간선 노드 전정 사이클 검출

Cycle Detection Using Single Edge Node Pruning

이상운*

Sang-Un Lee*

요약 본 논문은 단일 링크드 리스트의 사이클을 검출하는데 특화된 Floyd의 거북이와 토끼 경주법이 다중 입력, 다중 출력을 갖는 무 방향 그래프, 방향 그래프, 트리 등에 대해서는 사이클 검출 실패의 단점을 보완한 알고리즘을 제안하였다. 제안된 알고리즘은 단순히 단일 간선을 갖는 원천(source)과 싱크(sink)를 가지치기하는 단일 간선 노드 전정 사이클 검출 방법을 적용하였다. 제안된 알고리즘을 다양한 리스트, 무 방향 그래프, 방향 그래프, 트리 등에 적용한 결과 모든 경우에 대해 사이클을 검출하는데 성공하였다. 따라서 제안된 알고리즘은 사이클 검출 분야에서 가장 단순하고 빠른 장점을 갖고 있다.

Abstract This paper proposes an algorithm that remedy Floyd's the tortoise and the hare algorithm (THA) shortcomings which is specialized in singly linked list (SLL), so this algorithm fails to detect the cycle in undirected graph, digraph, and tree with multiple inputs or outputs. The proposed algorithm simply pruning the source and sink with only one edge using cycle detection of single edge node pruning. As a result of the experimental of various list, undirected graph, digraph, and tree, the proposed algorithm can be successively detect the cycle all of them. Thus, the proposed algorithm has the simplest and fastest advantage in the field of cycle detection.

Key Words : the tortoise and the hare, the teleporting turtle, depth-first search, source, sink, pruning

1. 서 론

사이클 검출(cycle detection)은 이산수학분야에서 난수발생기, 암호학의 해시함수, 정수론, 컴퓨터 프로그램의 무한루프 등의 품질시험, DBMS의 트랜잭션 관리의 교착상태(deadlock) 검출, 링크드 리스트의 모양분석(shape analysis) 등에 적용되고 있다.

$\alpha^\gamma \equiv \beta \pmod{n}$ 에서 γ 을 찾는 이산대수 문제에 있어서

사이클을 이용하는 Pollard의 Rho^[1] 방법 등이 있다. 또한, 단일 링크드 리스트(singly linked list, SLL)에서의 사이클을 검출하는 가장 널리 알려진 방법으로는 Floyd의 거북이와 토끼(the tortoise and the hare) 방법(THA)^[2]과 Brent의 이동하는 거북이(the teleporting turtle) 방법(TTA)^[3]이 있다.

THA는 SLL에 최적화된 방법으로, 일반적인 트리, 무 방향 그래프와 방향그래프에 대해서는 적합하지 않다.

*정회원, 강릉원주대학교 과학기술대학 멀티미디어공학과
접수일자 2023년 10월 21일, 수정완료 2024년 1월 8일
게재확정일자 2024년 2월 9일

Received: 21 October, 2023 / Revised: 8 January, 2024 /
Accepted: 9 February, 2024

*Corresponding Author: sulee@gwnu.ac.kr

Dept. of Multimedia Eng., Gangneung-Wonju National
University, Korea

따라서 이들에 대해서는 깊이우선 법(depth-first search, DFS)이 적용되고 있다.^[4] 이외에도 Rungta, et al.^[5]와 Haeupler et al.^[6] 등이 있다. Agarwal wt al.^[7]은 다중 스레드 프로그램의 교착상태(deadlock)를 검출하는데 이용하였으며, Lee^[8]는 이산대수 문제를 푸는 방법에 적용하였다.

본 논문에서는 리스트, 트리, 무 방향 그래프, 방향 그래프 등 모든 경우에 대해 사이클을 찾을 수 있는 방법을 제안한다.

2장에서는 THA와 TTA를 고찰한다. 3장에서는 단일 간선을 가진 source와 sink를 삭제하는 사이클 검출 방법을 제안한다. 4장에서는 제안된 알고리즘의 성능을 검증하여 본다.

II. 대표적인 사이클 검출법

컴퓨터과학 분야에서, 사이클 검출 또는 사이클 찾기(cycle finding)는 반복적인 함수 값의 일련의 순서에서 사이클을 찾는 알고리즘적인 문제이다^[9].

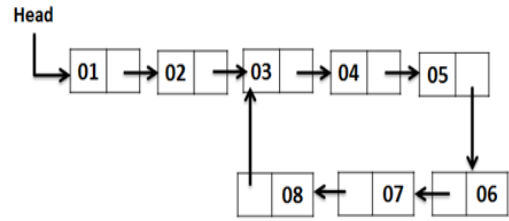
어떤 함수 f 는 유한집합 S 로 구성되어 있으며, 초기치 x_0 로 하여 식 (1)의 반복적인 함수의 일련의 순서로 사상되고 있다.

$$x_0, x_1 = f(x_0), x_2 = f(x_1), \dots, x_i = f(x_{i-1}), \dots \quad (1)$$

여기서, $i \neq j$ 인 $x_i = x_j$ 로 동일한 값이 발생할 수 있으며, 이 경우 사이클이 존재한다고 한다. 또한, x_0 부터 시작하여 사이클이 형성되는 가장 작은 수인 첫 번째 위치까지를 μ , 사이클의 길이를 λ 라 한다.

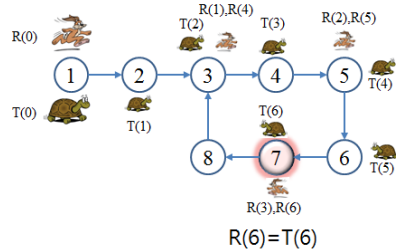
THA^[2]는 토끼와 거북이가 x_0 에서 출발하여 거북이($T(i)$)는 보폭 1로, 토끼($R(i)$)는 보폭 2로 경주를 하면서 $T(i) = R(i)$ 로 동일한 값을 얻으면 사이클이 검출되었다고 하며, 토끼가 일련의 유한함수 끝까지 도달하여도 거북이와 동일한 값을 얻지 못하면 사이클 검출에 실패(사이클이 없음)하였다고 한다.

TTA^[3]는 거북이가 $2^k, (k=0,1,2,\dots)$ 일 때 위치를 이동시키면서 토끼와 거북이가 동일한 값 $T(i) = R(i)$ 이면 사이클을 검출하는 방법으로 THA에 비해 보다 빠른 방법으로 알려져 있다.

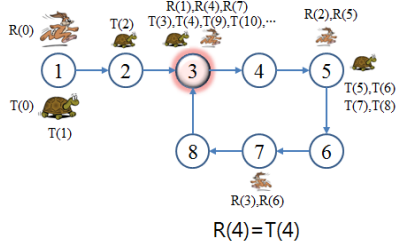


(a) G_1 SLL

k	횟수	THA		TTA	
		거북이	토끼	거북이	토끼
0	1	2	3	1	3
0→1	2	3	5	1	5
1	3	4	7	3	7
1→2	4	5	3	3	3
2	5	6	5	5	5
2	6	7	7	5	7
2	7	8	3	5	3
2→3	8	3	5	5	5
3	9	4	7	3	7
3	10	5	3	3	3
3	11	6	5	3	5



(b) THA



(c) TTA

그림 1. G_1 의 THA와 TTA
Fig. 1. THA and TTA for G_1

III. 단일 간선 노드 전정 알고리즘

본장에서 제안하는 방법은 그림 2와 같이 사이클이 존재하려면 어느 한 노드의 유입과 유출이 존재해야 하는 단순한 규칙에 기반하고 있다. 따라서 단일 간선(single edge)을 갖는 source($n \in x, n \notin f(x)$)와 sink($n \notin x, n \in f(x)$)는 삭제(또는 전정)된다.

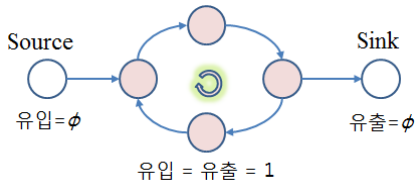


그림 2. 사이클 검출 규칙
 Fig. 2. Rule of cycle detection

제안된 알고리즘을 단일 간선 노드 전정 알고리즘 (single edge node pruning algorithm, SENPA)라 하며, 다음과 같이 수행된다.

```

Input :  $x_i, f(x_i)$ 
 $x_i = f(x_i)$  : 삭제 (재귀)
while  $n \in x, n \notin f(x)$  or  $n \notin x, n \in f(x)$ 
    if  $n \in x, n \notin f(x)$  then
        delete  $x_i, f(x_i)$  for  $x_i = n$ 
    if  $n \notin x, n \in f(x)$  then /* 방향그래프, 리스트 */
        delete  $x_i, f(x_i)$  for  $f(x_i) = n$ 
    if  $n \notin x, n \in f(x)$  and  $f(n) = 1$  then /* 무방향
    그래프, 트리 :  $f(n)$ : 빈도수,  $n$ 이 2회 이상 존
    재시 삭제 안함 */
        delete  $x_i, f(x_i)$  for  $f(x_i) = n$ 
    
```

do

그림 1의 G_1 에 대해 제안된 SENPA를 적용한 결과는 그림 3과 같다. 먼저 $\text{source}(n \in x, n \notin f(x))$ 인 “1” 노드가 제거되고, 다음으로 $\text{source}(n \in x, n \notin f(x))$ 인 “2” 노드가 삭제된다. 남은 3-4-5-6-7-8-3은 모두 $n \in x, n \in f(x)$ 로 사이클임을 알 수 있다.

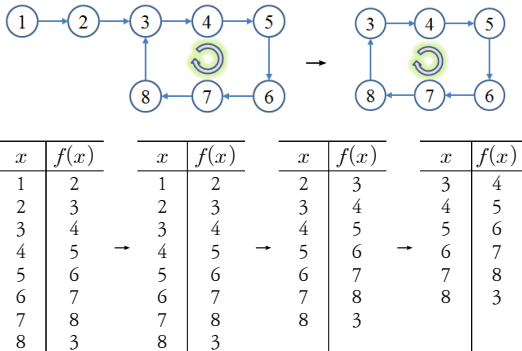
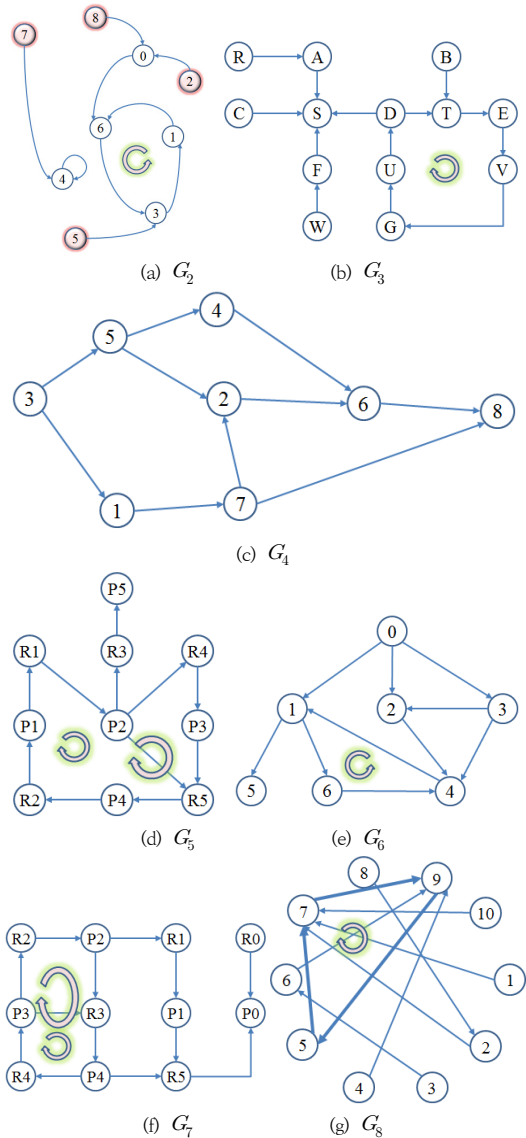
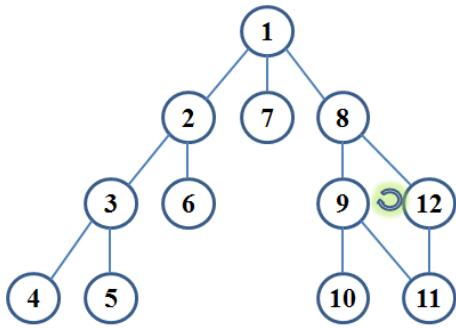


그림 3. G_1 의 단일 간선 노드 삭제 알고리즘
 Fig. 3. SENPA for G_1

IV. 적용 및 결과분석

본 논문에서 제안된 SENPA의 알고리즘 적합성을 검증하기 위해 적용된 실험 데이터는 그림 4에 제시되어 있다. 이들 데이터는 인터넷 상에서 Google 검색기로 검색된 데이터들이다.





(h) G_0

그림 4. 실험 데이터
Fig. 4. Experimental data

그림 4의 실험데이터에 대해 제안된 SENPA를 적용한 결과는 그림 5에 제시하였다.

x	$f(x)$	x	$f(x)$	x	$f(x)$	x	$f(x)$
0	6	0	6	0	6	1	6
1	6	1	6	1	6	3	1
2	0	2	0	3	1	6	3
3	1	3	1	6	3		
4	4	4	4				
5	3	5	3				
6	3	6	3				
7	4	7	4				
8	0	8	0				

cycle : 1-6-3-1

(a) G_2

x	$f(x)$	x	$f(x)$	x	$f(x)$
R	A	R	A	D	T
A	S	A	S	T	E
C	S	C	S	E	V
W	F	W	F	V	G
F	S	F	S	G	U
D	S	D	S	U	D
D	T	D	T		
B	T	B	T		
T	E	T	E		
E	V	E	V		
V	G	V	G		
G	U	G	U		
U	D	U	D		

cycle : D-T-E-V-G-U-D

(b) G_3

x	$f(x)$	x	$f(x)$	x	$f(x)$	x	$f(x)$
1	7	1	7	1	7		
2	6	2	6	2	6		
3	1	3	1	4	6		
3	5	3	5	5	2		
4	6	4	6	5	4		
5	2	5	2	7	2		
5	4	5	4				
6	8	6	8				
7	2	7	2				
7	8	7	8				

cycle : \varnothing

(c) G_4

x	$f(x)$	x	$f(x)$	x	$f(x)$	x	$f(x)$
P2	R3	P2	R3	P2	R3	P2	R4
R3	P5	R3	P5	P2	R4	P2	R5
P2	R4	P2	R4	P2	R5	R4	P3
P2	R5	P2	R5	R4	P3	P3	R5
R4	P3	R4	P3	P3	R5	R5	P4
P3	R5	P3	R5	R5	P4	P4	R2
R5	P4	R5	P4	R2	P4	R2	P1
P4	R2	P4	R2	R2	P1	P1	R1
R2	P1	R2	P1	P1	R1	R1	P2
P1	R1	P1	R1	R1	P2		
R1	P2	R1	P2				

cycle : R1-P2-R4-P3-R5-P4-R2-P1-R1
R1-P2-R5-P4-R2-P1-R1

(d) G_5

x	$f(x)$	x	$f(x)$	x	$f(x)$	x	$f(x)$	x	$f(x)$
0	1	0	1	1	6	1	6	1	6
0	2	0	2	6	4	6	4	6	4
0	3	0	3	4	1	4	1	4	1
1	5	1	5	2	4	2	4		
1	6	1	6	3	2	3	2		
6	4	6	4	3	4				
4	1	4	1						
2	4	2	4						
3	2	3	2						
3	4	3	4						

cycle : 1-6-4-1

(e) G_6

x	$f(x)$	x	$f(x)$	x	$f(x)$	x	$f(x)$	x	$f(x)$
R2	P2	R2	P2	R2	P2	R2	P2	R2	P2
P2	R1	P2	R1	P2	R1	P2	R1	P2	R1
P2	R3	P2	R3	P2	R3	P2	R3	P2	R3
R3	P4	R3	P4	R3	P4	R3	P4	R3	P4
P4	R4	P4	R4	P4	R4	P4	R4	P4	R4
P4	R5	P4	R5	P4	R5	P4	R5	P4	R5
R4	P3	R4	P3	R4	P3	R4	P3	R4	P3
P3	R3	P3	R3	P3	R3	P3	R3	P3	R3
P3	R2	P3	R2	P3	R2	P3	R2	P3	R2
R1	P1	R1	P1	R1	P1	R1	P1	R1	P1
P1	R5	P1	R5	P1	R5				
R0	P0	R0	P0						
R5	P0	R5	P0						

cycle : R2-P2-R3-P4-R4-P3-R2
P3-R3-P4-R4-P3

(f) G_7

x	$f(x)$	x	$f(x)$	x	$f(x)$	x	$f(x)$
1	7	1	7	2	7	5	7
2	7	2	7	5	7	7	9
3	6	3	6	6	9	9	5
4	9	4	9	7	9		
5	7	5	7	9	5		
6	9	6	9				
7	9	7	9				
8	2	8	2				
9	5	9	5				

cycle : 5-7-9-5

(g) G_8

x	$f(x)$	x	$f(x)$	x	$f(x)$	x	$f(x)$	x	$f(x)$
1	2	1	2	1	2	1	2	1	8
1	7	1	7	1	8	1	8	8	9
1	8	1	8	2	3	8	9	8	12
2	3	2	3	8	9	8	12	9	11
2	6	2	6	8	12	9	11	12	11
3	4	3	4	9	11	12	11		
3	5	3	5	12	11				
8	9	8	9						
8	12	8	12						
9	10	9	10						
9	11	9	11						
12	11	12	11						

cycle : 8-12-11-9-8
 (h) G_9

그림 5. 실험 데이터에 대한 SENPA
 Fig. 5. SENPA for Experimental data

본 논문에서 거론된 실험 데이터들에 대해 THA와 SENPA의 성능(수행횟수)을 비교한 결과는 표 1에 요약되어 있다.

THA는 G_1, G_4, G_8 을 제외한 데이터에 대해서는 실패할 가능성이 매우 높음을 알 수 있다. 반면에 SENPA는 모든 데이터에 대해 정확하면서도 빠르게 사이클을 검출할 수 있음을 알 수 있다. 단지 차이점은 THA는 수행 복잡도 $O(n)$, 메모리 요구량 $O(1)$ 인데 비해 SENPA는 수행 복잡도 $O(n)$, 메모리 요구량 $O(n)$ 으로 메모리 요구량에서 차이가 있을 뿐이다.

표 1. 알고리즘 수행횟수 비교
 Table 1. Compare with algorithm execution times

데이터	알고리즘 수행 횟수	
	THA	SENPA
G_1	6회	2
G_2	7: 실패 8: 3회	2
G_3	R, C, W : 실패 B: 7회	1
G_4	3-5-4-6-8: 2회 3-1-7-2-6-8: 3회	3
G_5	R1-P2-R3-P5 : 실패 R1-P2-R4-P3-R5-P4-R2-P1-R1: 9회	2
G_6	0-1-5 : 실패 0-2-4-1-5 : 실패 0-2-4-1-6-4 : 실패 0-2-4-1-6-4 : 4회	3
G_7	R0 : 실패 R2-P2-R1-P1-R5-P0 : 실패 R2-P2-R3-P4-R5-P0 : 실패 R2-P2-R3-P4-R4-P3-R3 : 5회	4
G_8	8-2-7-9-5-7 : 8회	2
G_9	1, ..., 4, 5, 6, 7, 10 : 실패 1-8-12-11-9-8 : 5회	4

V. 결론

Floyd의 THA는 SLL에 특화된 사이클 검출 기법으로 출발지점과 종료지점이 2개 이상인 경우에는 사이클 검출에 실패할 수 있다. 반면에 본 논문에서는 리스트, 무방향 그래프, 방향그래프, 트리 등의 일반적인 범용성을 가진 사이클 검출 알고리즘을 제안하였다.

제안된 알고리즘은 범용성을 가지도록하기 위해, Floyd의 토끼와 거북이 경주 기법 대신, 단일간선을 가지는 source와 sink를 가지치기(삭제)하는 간단한 방법을 제시하였다.

제안된 알고리즘을 다양한 경우인 리스트, 무방향 그래프, 방향그래프, 트리 등에 적용한 결과 THA로는 사이클 검출에 실패하는 경우에 대해서도 제안된 SENPA는 사이클을 정확하게 검출할 수 있음을 보였다.

따라서 제안된 SENPA를 사이클 검출에 있어서 범용성을 가진 알고리즘으로 활용할 수 있을 것이다.

References

- [1] J. M. Pollard, "Monte Carlo Methods for Index Computation (mod p)," Mathematics of Computation, Vol. 32, No. 143, pp. 918-924, doi:10.2307/2006496, Jul. 1978.
- [2] R. W. Floyd, "Non-deterministic Algorithms," Journal of ACM, Vol. 14, No. 4, pp. 636-644, Oct. 1967. doi:10.1145/321420.321422
- [3] R. P. Brent, "An Improved Monte Carlo Factorization Algorithm," Bit Numerical Mathematics (BIT), Vol. 20, No. 2, pp. 176-184, Jun. 1980. doi:10.1007/BF01933190
- [4] A. Kamil, "Graph Algorithms," CS61B, Spring 2003 Discussion #13, UC Berkeley, <https://web.eecs.umich.edu/~akamil/teaching/sp03/041403.pdf>, Apr. 2013.
- [5] S. Rungta, S. Srivastava, U. S. Yadav, and R. Rastogi, "A Methodology to Find the Cycle in a Directed Graph Using Linked List," Bharati Vidyapeeth's Institute of Computer Applications and Management (BVICAM)'s International Journal of Information Technology (BIJIT), Vol. 6, No. 2, pp. 743-749, Dec. 2014.
- [6] B. Haeupler, T. Kavitha, R. Mathew, S. Hasen, and R. Tarjan, "Incremental Cycle Detection, Topological Ordering, and Strong Component Maintenance," ACM Transactions on Algorithms, Vol. 8, No. 1, pp. 3:1-3:33, Jan. 2012. doi: 10.1145/2071379.2071382
- [7] R. Agarwal, S. Bensalem, E. Farchi, K. Havelund, Y. Nir-Buchbinder, S. D. Stoller, S. Ur, and L. Wang,

"Detection of Deadlock Potentials in Multithreaded Programs," IBM Journal of Research & Development, Vol. 54 No. 5, pp. 3:1-3:15, Sep. 2010.
doi: 10.1147/JRD.2010.2060276

- [8] S. U. Lee, "Cycle Detection in Discrete Logarithm Using a Queue," The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol. 17, No. 3, pp. 1-7, Jun. 2017.
doi.: 10.7236/IIBC.2017.17.3.1
- [9] Wikipedia, "Cycle Detection," http://en.wikipedia.org/wiki/Cycle_detection, Wikimedia Foundation, Inc, Retrieved Oct. 2023.

저 자 소 개

이 상 운(정회원)



- 1987년 : 한국항공대학교 항공전자공학과 (학사)
- 1997년 : 경상대학교 컴퓨터과학과 (석사)
- 2001년 : 경상대학교 컴퓨터과학과 (박사)
- 2003년 : 강원도립대학 컴퓨터응용과 전임강사
- 2004년 ~ 2007.2 : 국립 원주대학 여성교양과 조교수
- 2007.3 ~ 2015.3 : 강릉원주대학교 멀티미디어공학과 부교수
- 2015.4 ~ 현재 : 강릉원주대학교 멀티미디어공학과 정교수
- 관심분야 : 소프트웨어 프로젝트 관리, 개발 방법론, 분석과 설계 방법론, 시험 및 품질보증, 소프트웨어 신뢰성, 인공지능과 빅데이터분석, 최적화 알고리즘
- e-mail : sulee@gwnu.ac.kr