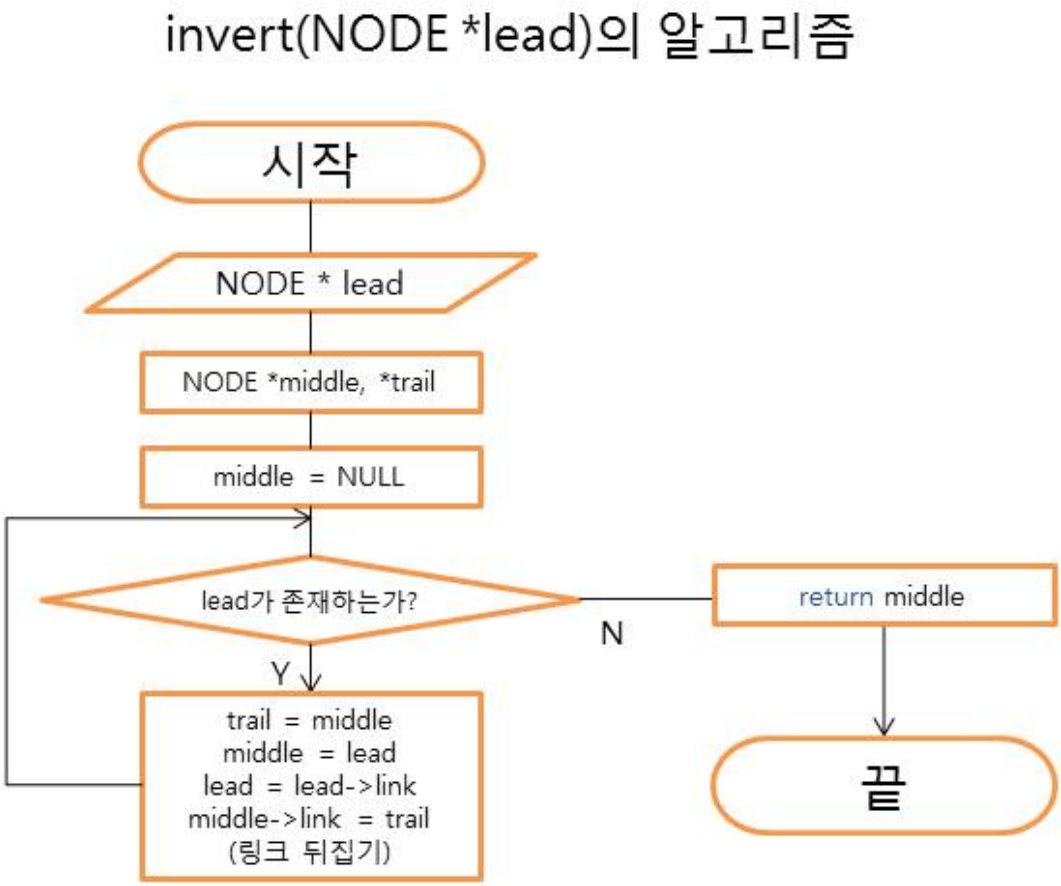
5. 연결리스트를 이용한 뒤집기(invert) 연산

21410785 황 희

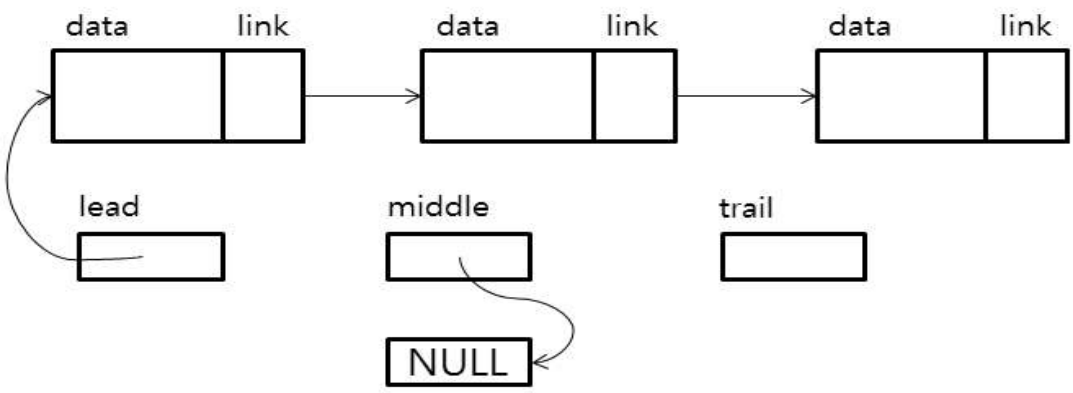
1. 개요

- 뒤집기 invert(NODE *lead) 의 알고리즘

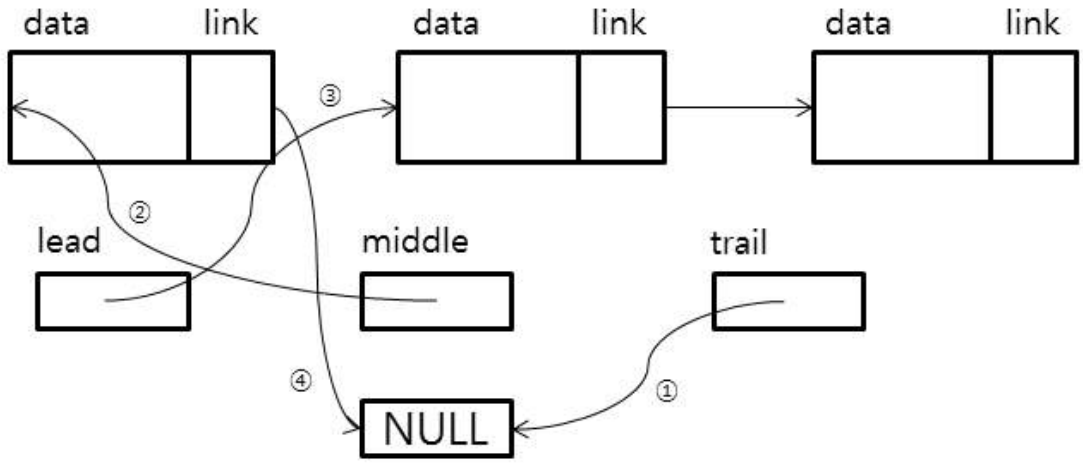


- 알고리즘 진행도

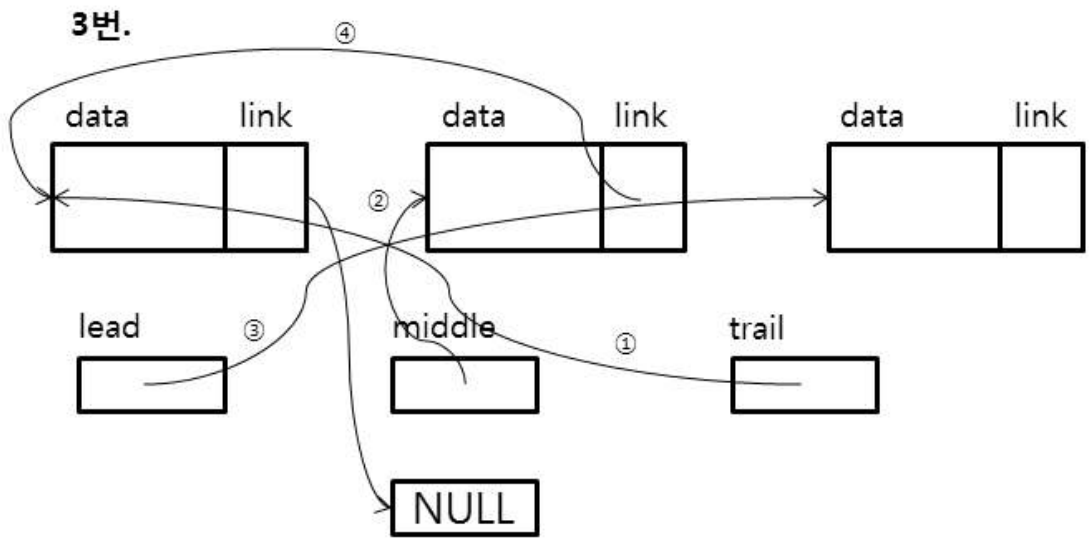
1번.



2번.



3번.



invert함수는 연결리스트를 역으로 뒤집어주는 역할을 하는 함수이다. 즉 노드의 순서를 모두 역순으로 돌린다. 먼저 매개변수로 연결리스트의 제일 처음노드를 나타내는 lead를 받는다. 다음, invert에 사용될 노드포인터 middle과 trail을 선언한다. middle에는 중간노드를 가리키는 포인터가 될 것이고, trail은 middle 전에 있는 노드를 가리킬 것이다. middle은 NULL의 주소를 주고, middle은 lead의 주소를 준다. 다음 lead는 lead->link로 다음노드로 이동하게 한다. 이동 한 노드에서의 link를 lead가 방금 지나왔던 노드의 주소, 즉 trail를 가리키게 하면 노드에서 노드를 연결하던 화살표가 반대로 간다. 이런식으로 lead가 NULL일때까지 가면 제일 뒤집어진 노드의 head부분은 middle이 될 것이므로 invert함수는 middle을 반환한다.

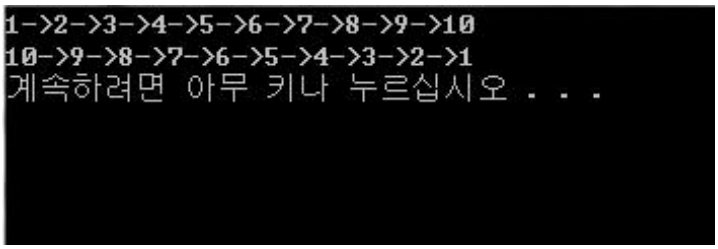
2. 본문

- 코드(설명은 주석으로 대체)

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
typedef struct node // 노드 구성
{
	int data; // 데이터
	struct node * link; // 노드를 연결해주는 링크
}NODE;
void addNode(NODE ** head, int cof, int exp) // 노드를 추가하는 함수
{
	NODE * p = *head; // head의 주소값을 p에 저장
	NODE * temp = (NODE*)malloc(sizeof(NODE));
	// 연결리스트 끝에 연결시킬 temp노드 생성
	temp->link = NULL; // temp노드가 리스트의 제일 끝임을 알려줌
	temp->data = data; // temp노드의 data를 입력받은 데이터로 지정
	if (*head == NULL) // 노드가 하나도 없다면 (제일 처음노드생성)
	{
		*head = temp; // head가 가리키는 값을 temp로 지정
		return; // 끝
	}
	// 노드가 이미 있다면
	while (p->link) p = p->link;
	// p가 가리키는 노드가 있다면 가리키는 노드가 없을때까지 노드를 이동
	p->link = temp; // 마지막 노드 p가 temp를 가리키도록 설정
}
NODE* invert(NODE* lead) // 연결리스트의 순서를 역순으로 뒤집는 함수
{
	NODE *middle, *trail; // invert에 사용 될 노드포인터 선언
	middle = NULL; // middle을 NULL로 정의
	while (lead) {
		trail = middle; // 꼬리부분이 middle을 가리키게 함
		middle = lead; // middle이 lead를 가리키게 함
		lead = lead->link; // lead를 다음노드로 이동
		middle->link = trail; //링크 뒤집기
	}
	return middle; //뒤집어진 리스트의 헤드
}
void showList(NODE * head) // 리스트를 출력하는 함수
{
	NODE * p = head;
	while (p)
	{
		printf("%d%s", p->data, p->link ? "->" : "");
		p = p->link;
```

```
        }
}
int main()
{
        int temp;
        NODE * x = NULL, *y = NULL;
        int temp;
        NODE * x = NULL, *y = NULL;
        for (int i = 0; i < 10; i++) {
                addNode(&x, i + 1);
        }
        showList(x);
        y = invert(x);
        showList(y);
}
```

- 결과창



먼저, 1에서부터 10까지를 연결리스트로 생성하여 출력하였다. 그 다음 invert함수를 사용하여 뒤집어진 head부분을 y에 넣어주었다. y리스트를 출력하는 함수를 넣으니 연결리스트가 뒤집어진 것을 확인 할 수 있다.

3. 결론

invert와 같이 노드의 순서를 뒤집는 등의 함수에는 무조건 임시 노드포인터가 하나 필요하고, 노드와 그 다음노드를 가리킬 노드포인터가 필요하다. invert함수에서는 뒤집을 리스트의 첫 번째 노드포인터를 lead라 하고, 뒤집었을 때 제일 첫 노드를 가리킬 노드포인터를 middle이라 하고 그 앞에 노드를 가리키는 포인터를 trail이라고 했다. 연결리스트를 invert와 같이 뒤집거나 수정할 때에는 link만 무작정 바꿔서는 안 되며, 임시 노드포인터를 지정하여 수정할 link를 보관하고 있어야 한다.

2017-12-10

황 희