



과목명	프로그래밍 언어개론[00]
담당교수님	조은선 교수님
이름	정성욱
학번	201404376

(1) RecursionLinkedList에 대한 해결 방법:

들어가기전 저는 짜여진 틀을 적극적으로 활용하였습니다. 매 코드를 짤 때마다 이

```
private static class Node {
    char item;
    Node next;
    Node(char element, Node next) {
        this.item = element;
        this.next = next;
    }
}
```

Node의 구조를 확인했었습니다.

과제 (1) 주어진 Node x의 마지막으로 연결된 Node의 다음으로 새롭게 생성된 노드를 연결

private void linkLast(char element, Node x)

My solution: 목적은 x노드의 마지막에 element를 가진 노드를 하나 더 추가하는 것입니다.

우선 재귀를 써야하기 때문에 마지막인지 아닌지 판단하는 분기문으로 if문을 추가하고 다음에 연결된 노드 next가 null일 경우 마지막이라 판단하고 linkNext라는 노드를 생성하는 메소드를 호출합니다. 그리고 현재 마지막 노드와 element를 줍니다.

그렇지않고 계속 재귀를 수행해야하는 경우는, x = x.next로 다음 노드로 옮겨준후 element랑 x를 그대로 매개변수로 linkLast를 수행합니다.

과제 (2) x노드에서 index만큼 떨어진 Node 반환

private Node node(int index, Node x)

My solution: 목적은 x노드에서 index만큼 떨어진 Node를 반환하는 것입니다. 즉 이 index는 하나의 거리라고도 칠 수 있습니다. 함수를 한번 씩 돌아갈 때마다 1씩 줄어드는 것입니다. 그리고 이 index는 재귀 호출을 몇 번 할 것인지에 대한 정보를 담고있습니다. 즉 index가 0인지 확인하고 0이면 탈출하도록 짜면 그만 인 것입니다. 물론 재귀를 한번 할 때마다 1씩 줄어들게 하는 것입니다.. 다시 정리하자면, 함수의 끝엔 분기문 index==0이면 x를 리턴하는 것을 만들고, index가 0이 아니라면 1을 줄인뒤에 다시 재귀호출을 하는 것입니다.

과제 (3) 노드로부터 끝까지의 리스트의 노드 갯수 반환

private int length(Node x)

My solution: 기본적인 구동원리는 위의 node와 같습니다. 그러나 몇 가지 차이점이려면 리턴값이 int라는 것과 매개변수가 x하나인 것입니다. 이것 또한 생각보다 간단히 풀수있는데, 바로 재귀를 호출할때마다 (현재의 노드 길이)1을 더하고 재귀호출의 매개변수로 x.next(다음은 가르키는) 것을 주는 것입니다. 그리고 분기문으로 현재 x가 null이면 아무것도 없는 즉, 노드의 끝부분보다 더 지나친 것이므로 0으로 리턴하게 합니다.(각 하나의 노드당 1씩 가지고 있으므로 호출하면서 자기 길이를 다 더하는 원리임)

과제 (4) 노드로부터 시작하는 리스트의 내용 반환

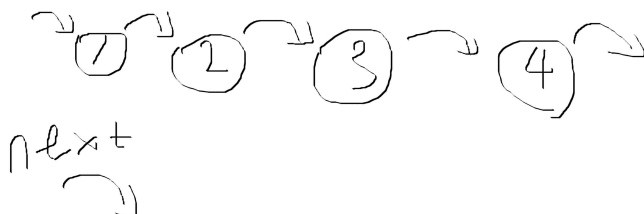
private String toString(Node x)

My solution: 기본적인 구동원리는 위의 함수들과 같습니다. x.next값이 null인 즉, 노드의 끝이면 일반적인 String만을 반환하는 분기문을 추가하고 끝이 아니라면, return을 하면서 x.item + 재귀호출(매개변수로 x.next)를 주는 식으로했습니다. 그러면 자동적으로 현재의 문자를 리턴하면서 마지막의 if문의 조건에 맞으면 재귀를 호출하지않고 현재의 문자만 리턴합니다.

추가 과제 (5) 현재 노드의 이전 노드부터 리스트의 끝까지를 거꾸로 만들

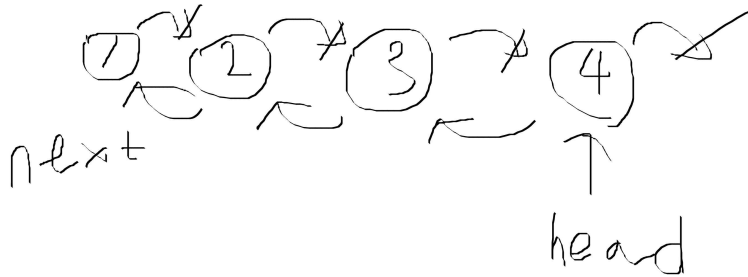
private void reverse(Node x, Node pred)

그림이 없으면 제 필력으로 설명이 어려울꺼같아서 첨부합니다.



우선 노드 1,2,3,4가 있고, 화살표는 next를 가르킵니다. 우리의 목표는 저 화살표들을 전부다 반대 방향으로 놓는 것입니다. 그러나 현재의 다음을 가르키는 next를 매개변수 pred로 바로 바꾸면 다음으로 진행하기가 정말 어려워지므로 새로운 temp노드를 하나 만들어 temp = x.next로 다음을 가르키게 한다음 next를 pred로 바꿉니다. 그리고 다음 재귀호출을 할 때(temp(옛날의 next가 가르키던 값),pred로 x(현재의값))으로 매개변수를 줍니다. 그리고

분기조건인 `x.next==null`을 주고 `x.next=pred`로 주고 헤드를 `head=x`로 옮기고 그냥 리턴을 하게되면



자연스럽게 이런모양이 됩니다.

추가 과제 (6) 두 리스트를 합침 (A + B)

private void addAll(Node x, Node y)

우선 여기서는 매개변수 x의 뒷부분에 y의 모든 것을 순서대로 이어붙인다는 것입니다. 이말은 무슨말이냐? 시작을 x의 가장 뒷부분까지 가야지 시작이므로 if문에 조건을 걸어서 x가 가장 마지막이 아니면, `x.next,y`를 매개변수로 재귀호출을 시행합니다. 그리고 이제부터가 시작인데 `elseif`를 붙여서 y역시 `y.next`가 null이 아니면 `x,y.next`를 시행하는 재귀호출을 합니다. 하지만 그전에! y의 값을 x의 뒷부분에 넣어줘야하므로, `linkLast` 메소드를 호출합니다 `y.item`과 `x`를 매개변수로 줍니다. 그러면 이 함수가 끝나고 새로 시작할때는 x가 마지막이 아니므로 다시 맨윗 조건에 따라 x를 한칸 이동하고 다시 y를 추가하고 이런식으로 진행하다보면 y가 마지막을 맞이합니다 `x.next도null` `y.next도 null`이므로 y의 마지막에 다다른 셈입니다. 이때는 그저 `linkLast(y.item,x)` 이렇게 하던대로 매개변수 넣고 `return` ;로 재귀호출을 끝내줍니다.

(3)느낀 점

처음부터 짜라했으면 조금 힘들었을 것 같습니다. 대부분 링크드 리스트나 틀을 만들어 주셨기에, `iterator`없이 만들수있었습니다. 재밌었습니다.

(4)구현 결과

```
[Iloveprogramminglanguage]
[egeugnalgnimmargorpevolI]
[geugnalgnimmargorpevolI]
[LOLgeugnalgnimmargorpevolI]
```