

자료구조응용

3. 회문(Palindrome) 검사, 승자 트리(Winner Tree)

1. 회문이란, “가다가”, “기러기”, “refer”, “level”, “rotator” 등과 같이 왼쪽에서 오른쪽으로, 오른쪽에서 왼쪽으로 읽어도 같은 문장을 말합니다.

1-1. 배열을 이용한 회문검사

(1) 실행순서

- ① 키보드로부터 문자열을 입력받음
- ② 반복문을 이용하여 주어진 문자열이 회문인지 검사하는 `int isPalindrome(char* str)` 함수를 작성하시오. `isPalindrome` 함수는 주어진 문자열 `str`이 회문이면 1을 회문이 아니면 0을 반환함.

실행 예)

```
C:\Windows\system32\cmd.exe
Enter the string to check palindrome: abcdef
abcdef is not palindrome
계속하려면 아무 키나 누르십시오 . . .
```

```
C:\Windows\system32\cmd.exe
Enter the string to check palindrome: abcba
abcba is palindrome
계속하려면 아무 키나 누르십시오 . . .
```

1-2. 연결 리스트를 이용한 회문검사

본 실습을 위한 자료구조는 다음과 같습니다.

```
typedef struct node* tListNodePointer;
typedef struct node {
    char data;
    tListNodePointer link;
} tListNode;

tListNodePointer head = NULL;
```

(1) 실행순서

- ① 키보드로부터 문자열을 입력받음
- ② 주어진 문자열로부터 연결 리스트를 생성하는 `tListNodePointer initList(char* str)` 함수를 작성하시오. `initList` 함수는 주어진 문자열로부터 연결 리스트를 생성하여 연결 리스트

의 제일 첫 번째 노드를 가리키는 포인터를 반환함.

- ③ 생성한 연결 리스트로부터 주어진 문자열이 회문인지 검사하는 `int isPalindrome(tListNodePointer head)` 함수를 작성하시오. `isPalindrome` 함수는 주어진 연결 리스트가 회문이면 1을 회문이 아니면 0을 반환함.

(2) `int isPalindrome(tListNodePointer head)` 함수 세부 구현 사항

- ① 본 함수의 구현을 위해서는 입력으로 주어진 연결 리스트를 복사는 `copy` 함수를 구현해야 합니다. `copy` 함수는 `traverse` 함수를 수정하여 구현합니다.
- ② 본 함수의 구현을 위해서는 역변환인 `invert` 함수를 구현해야 합니다. `invert` 함수 호출 후 `head` 및 역변환 연결 리스트를 화면에 출력합니다.
- ③ `invert` 함수를 통해서 생성된 역변환된 연결 리스트와 `head`로 연결된 연결 리스트의 각 노드를 방문하면서 각 노드의 `data field`가 일치하는 검사합니다.(금일 수업한 연결 리스트 복습 수업자료의 `traverse` 함수를 수정하여 구현합니다.)

```
tListNodePointer invert(tListNodePointer lead)
{
    tListNodePointer middle, trail;
    middle = NULL;
    while (lead) {
        trail = middle;
        middle = lead;
        lead = lead->link;
        middle->link = trail;
    }
    return middle;
}
```

invert 함수

```
void traverse()
{
    tListNodePointer curr = head;
    while (curr) {
        printf("%s\n", curr->data.name);
        curr = curr->link;
    }
    return curr;
}
```

traverse 함수

실행 예)

```
C:\Windows\system32\cmd.exe
Enter the string to check palindrome: abdefg
initial linked list: a b d e f g
inverted linked list: g f e d b a
abdefg is not palindrome
계속하려면 아무 키나 누르십시오 . . .
```

```
C:\Windows\system32\cmd.exe
Enter the string to check palindrome: abdefgfedba
initial linked list: a b d e f g f e d b a
inverted linked list: a b d e f g f e d b a
abdefgfedba is palindrome
계속하려면 아무 키나 누르십시오 . . .
```

2. [승자트리를 이용한 정렬] 10개의 정수형 원소로 구성된 k 개의 레코드를 가지는 승자트리 (winner tree)의 초기생성 함수를 작성하여 정렬을 수행하라. 이때 k 는 2의 거듭승 (power of 2)임을 가정하라.

(1) 실행순서

① 난수생성을 위한 seed와 k 를 입력받는다.

② $k \times 10$ 의 2차원 배열을 생성한다.

③ 1~100 사이에서 발생시킨 k 개의 난수를 key로 사용하여 각 레코드에 k 부터 $k+9$ 까지의 숫자를 순서대로 배열에 저장한다.

※ 각 key는 중복 가능하다.

생성된 k 개의 난수($k=8$)

10차원 배열	10	9	20	6	8	9	90	17
	11	10	21	7	9	10	91	18
	12	11	22	8	10	11	92	19
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	19	18	29	15	17	18	99	26

$k = 8$ 일 때의 난수 생성 후 레코드의 예

- ④ ②에서 생성한 키 데이터에 대해 초기 승자트리를 구성한다. ※ `initWinner()`
- ※ winner tree 는 완전이진트리이며 노드 레벨에 따라 배열에 순차적으로 저장된다.
 - ※ winner tree 의 각 노드는 키 값(혹은 레코드)에 대한 포인터(혹은 인덱스)만을 가진다.
 - ※ `initWinner()`는 recursive postorder traversal 형태로 작성한다.
- ⑤ 승자트리를 사용한 정렬을 수행한다.
- ※ { 최소키를 sorted 배열에 저장 -> 최소키가 최종 승자로 선택된 레코드에는 다음 키 값으로 `nums` 배열의 값을 치환(레코드의 키 값을 다 소진한 경우에는 무한대를 의미하는 임의의 값으로 최소키를 치환) -> 승자트리를 재구성(`adjustWinner()`) } 이 과정을 $k * 10$ 번 반복함
 - ※ `adjust` 시, 치환된 키의 index --> parent index --> sibling index를 구할 수 있음. 치환된 키와 sibling 키의 비교를 루트 방향으로 수행함
- ⑥ 정렬된 결과배열의 인덱스 순서대로 키 값을 출력한다.

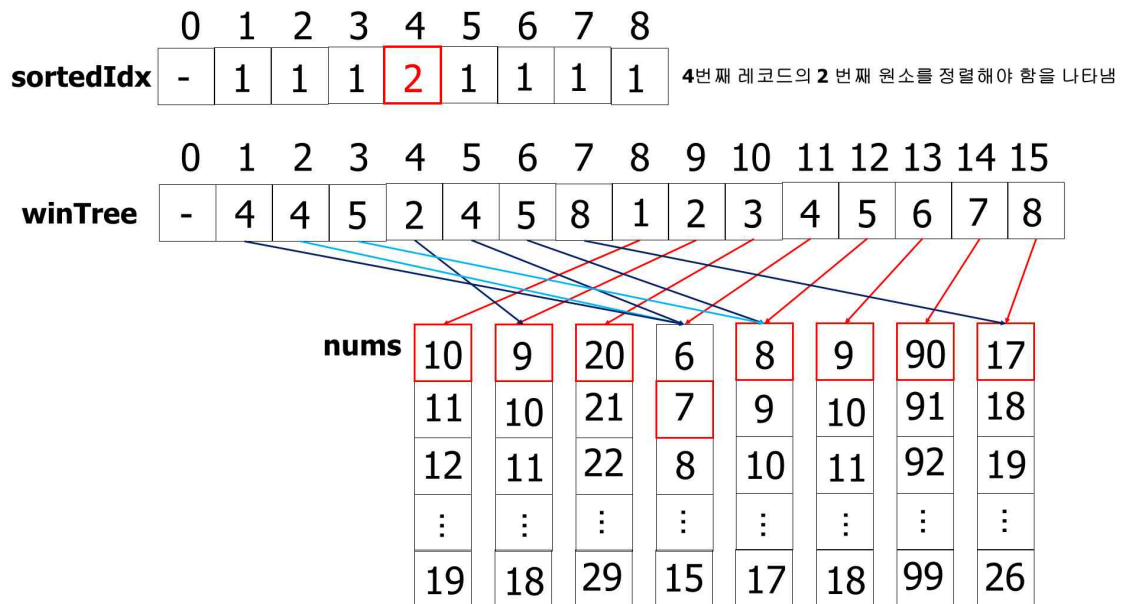
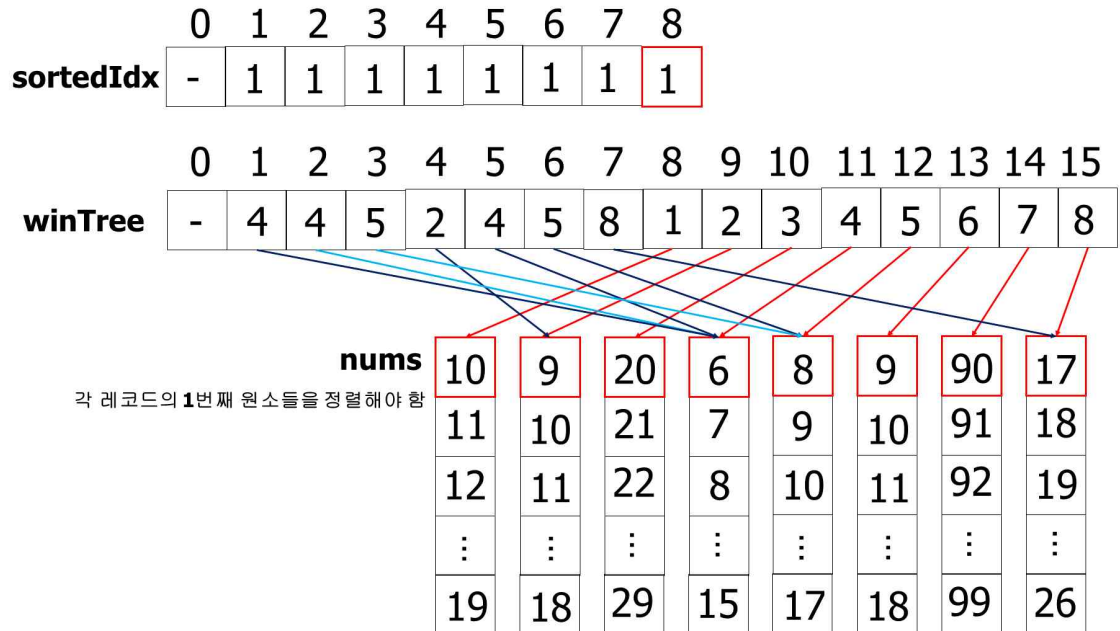
(2) 구현세부사항

① 함수정의 및 트리선언

```
#define MAX_SIZE 100
#define INF_NUM 10000

// min - winner tree
int nums[ MAX_SIZE+1 ][11] = { 0 };           // keys to sort
int winTree[ 2*MAX_SIZE ] = { NULL };         // winner tree
int sorted[ MAX_SIZE*10+1 ] = { 0 };          // sorted result
int initWinner(int cur, int k, int winTree[]);
void adjustWinner(int min, int k, int winTree[]);
void inorder(int root, int k, int winTree[]);
```

② k=8일 경우, 초기 min - winner tree 구성 예



sorted - 6

4번째 레코드의 key 값 6이 정렬되고 난 후의 sortedIdx, winTree, nums의 상태

(3) 실행 예

```
C:\Windows\system32\cmd.exe
```

```
<<<<<<<<< sorting with winner tree >>>>>>>>>>>>>>>>  
the number of keys ( 8, 16, or 32 as a power of 2 ) >> 8  
random numbers to use as key values (1 ~ 100)  
seed >> 1  
initial records:  
1-th records:  
42 43 44 45 46 47 48 49 50 51  
2-th records:  
68 69 70 71 72 73 74 75 76 77  
3-th records:  
35 36 37 38 39 40 41 42 43 44  
4-th records:  
1 2 3 4 5 6 7 8 9 10  
5-th records:  
70 71 72 73 74 75 76 77 78 79  
6-th records:  
25 26 27 28 29 30 31 32 33 34  
7-th records:  
79 80 81 82 83 84 85 86 87 88  
8-th records:  
59 60 61 62 63 64 65 66 67 68  
  
initialization of min-winner tree  
  
inorder traversal for min-winner tree  
  
sorting with min-winner tree...  
  
sorted result  
   1    2    3    4    5    6    7    8  
   9   10   25   26   27   28   29   30  
  31   32   33   34   35   36   37   38  
  39   40   41   42   42   43   43   44  
  44   45   46   47   48   49   50   51  
  59   60   61   62   63   64   65   66  
  67   68   68   69   70   70   71   71  
  72   72   73   73   74   74   75   75  
  76   76   77   77   78   79   79   80  
  81   82   83   84   85   86   87   88  
계속하려면 아무 키나 누르십시오 . . . _
```

```
C:\Windows\system32\cmd.exe
<<<<<<<<<< sorting with winner tree >>>>>>>>>>
the number of keys ( 8, 16, or 32 as a power of 2 ) >> 4
random numbers to use as key values (1 ~ 100)
seed >> 10
initial records:
1-th records:
72 73 74 75 76 77 78 79 80 81
2-th records:
100 101 102 103 104 105 106 107 108 109
3-th records:
73 74 75 76 77 78 79 80 81 82
4-th records:
95 96 97 98 99 100 101 102 103 104
initialization of min-winner tree
inorder traversal for min-winner tree
sorting with min-winner tree...
sorted result
72 73 73 74
74 75 75 76
76 77 77 78
78 79 79 80
80 81 81 82
95 96 97 98
99 100 100 101
101 102 102 103
103 104 104 105
106 107 108 109
계속하려면 아무 키나 누르십시오 . . .
```

■ 제출 형식

- 제출 : 12월 6일자 채점서버에 소스 업로드(12월 19일까지)
- 실행화면을 캡처하여 한글파일 보고서(DS 3_학번.hwp)에 추가 후 과제에 제출 (12월 19일까지)
- 각 소스파일에 주석처리 추가
“학번 이름”
“본인은 이 소스파일을 다른 사람의 소스를 복사하지 않고 직접 작성하였습니다.”

■ 주의

- **소스복사로는 실력향상을 기대할 수 없습니다!!!**
- 먼저 제출한 학생은 남은 시간 동안 자료구조 관련 개인학습을 하거나 동료를 도와 줄 것
- 수강생 끼리 서로 물어보고 논의를 해도 됨
- 채점서버에 제출하지 않는 경우 최종 점수의 0점으로 처리함
(사용법을 잘 모르겠다면 개인적으로 튜터나 TA를 찾아와서 물을 것)