

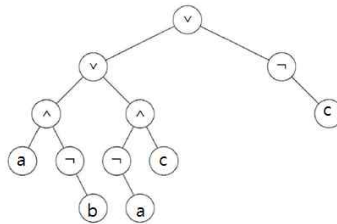
자료구조응용

13. Trees : 명제식 이진트리, heap

1. postfix expression의 명제식을 파일로부터 입력받아 이진트리를 구성하여 중위순회 (inorder traversal)한 결과를 화면에 출력하라.

(1) 입력파일, 이진트리 및 실행 예

- 입력파일(postfix.txt) : ab~&a~c&|c~|
- 피연산자(Operands) : 알파벳 소문자
- 연산자(Operators) : & | ~



```
C:\WINDOWS\system32\cmd.exe
input_string(postfix) : ab~&a~c&|c~|
creating its binary tree
inorder traversal      : a&~b|~a&c|~c
계속하려면 아무 키나 누르십시오 . . .
```

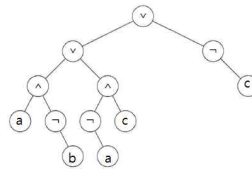
(2) 실행순서

- ① postfix expression의 명제식 (1번 문제의 활용가능)으로 부터 이진트리를 생성한다.
- ② 이진트리에 대한 중위 순회를 수행하여 명제식을 출력한다.

2. postfix expression의 명제식을 파일로부터 입력받는다. 이 명제식을 참으로 만들기 위해 입력 되는 변수에 대한 모든 경우의 값을 구하시오. 단 프로그램의 편의를 위해 입력되는 변수는 3개로 한다.

(1) 입력파일, 이진트리 및 실행 예

- 입력파일(postfix.txt) : ab~&a~c&|c~|
- 피연산자(Operands) : 알파벳 소문자
- 연산자(Operators) : & | ~



(2) 실행순서

- ① postfix expression의 명제식 (1번 문제의 활용가능)으로 부터 이진트리를 생성한다.
- ② 이진트리에 대한 후위 순회를 수행하여 명제식을 계산한다.

```
C:\WINDOWS\system32\cmd.exe
input string(postfix) : ab~&a~c&|c~|
find true condition
a=0, b=0, c=0
a=0, b=0, c=1
a=0, b=1, c=0
a=0, b=1, c=1
a=1, b=0, c=0
a=1, b=0, c=1
a=1, b=1, c=0
계속하려면 아무 키나 누르십시오 . . .
```

3. 다음 입력파일의 데이터를 사용하여 최대힙(Max Heap)에 대한 실습을 수행한다.

input.txt : 10 40 30 5 12 6 15 9 60

(1) 실행순서

① 파일입력을 받으면서 최대힙을 구성한다.

매 입력마다, 구성된 최대힙의 배열원소를 인덱스 순서대로 출력한다.

② 최대힙의 최대값을 연속으로 원소개수만큼 삭제한다.

매 삭제마다, 재구성된 최대힙의 배열원소를 인덱스 순서대로 출력한다.

(3) 구현 세부사항

```
#define MAX-ELEMENTS 200 /* maximum heap size+1 */
#define HEAP-FULL(n) (n == MAX-ELEMENTS-1)
#define HEAP-EMPTY(n) (!n)
typedef struct {
    int key;
    /* other fields */
} element;
element heap[MAX-ELEMENTS];
int n = 0;
```

```
void push(element item, int *n)
{ /* insert item into a max heap of current size *n */
    int i;
    if (HEAP-FULL(*n)){
        fprintf(stderr, "The heap is full. \n");
        exit(EXIT_FAILURE);
    }
    i = ++(*n);
    while ((i != 1) && (item.key > heap[i/2].key)) {
        heap[i] = heap[i/2];
        i /= 2;
    }
    heap[i] = item;
}
```

Program 5.13: Insertion into a max heap

```

element pop(int *n)
{
    /* delete element with the highest key from the heap */
    int parent, child;
    element item, temp;
    if (HEAP_EMPTY(*n)) {
        fprintf(stderr, "The heap is empty\n");
        exit(EXIT_FAILURE);
    }
    /* save value of the element with the highest key */
    item = heap[1];
    /* use last element in heap to adjust heap */
    temp = heap[(*n)--];
    parent = 1;
    child = 2;
    while (child <= *n) {
        /* find the larger child of the current parent */
        if ((child < *n) && (heap[child].key < heap[child+1].key))
            child++;
        if (temp.key >= heap[child].key) break;
        /* move to the next lower level */
        heap[parent] = heap[child];
        parent = child;
        child *= 2;
    }
    heap[parent] = temp;
    return item;
}

```

Program 5.14: Deletion from a max heap

(4) 실행 예

```

C:\WINDOWS\system32\cmd...
***** insertion into a max heap *****
10
40 10
40 10 30
40 10 30 5
40 12 30 5 10
40 12 30 5 10 6
40 12 30 5 10 6 15
40 12 30 9 10 6 15 5
60 40 30 12 10 6 15 5 9
***** deletion from a max heap *****
delet data = 60
40 12 30 9 10 6 15 5
delet data = 40
30 12 15 9 10 6 5
delet data = 30
15 12 6 9 10 5
delet data = 15
12 10 6 9 5
delet data = 12
10 9 6 5
delet data = 10
9 5 6
delet data = 9
6 5
delet data = 6
5
delet data = 5
계속하려면 아무 키나 누르십시오 . . .

```

■ 제출 형식

- 공학인증 시스템(ABEEK)에 과제를 올릴 때 제목:
 - 1차 제출: 학번 이름 DS-13(1), 2차 제출: 학번 이름 DS-13(2)
- 솔루션 이름 : DS-13
- 프로젝트 이름 : 1, 2, 3
- 실행화면을 캡처하여 한글파일에 추가 후 솔루션 폴더에 포함.
- 한글 파일명 : 학번_이름_실습결과.hwp
- 솔루션 폴더를 압축하여 게시판에 제출할 것.
- 솔루션 압축 파일 명:
 - 1차 제출: 학번_이름_DS_13(1).zip, 2차 제출: 학번_이름_DS_13(2).zip
- 제출은 2회 걸쳐 가능(수정 시간 기준으로 처리)