Data Structure

Week 10

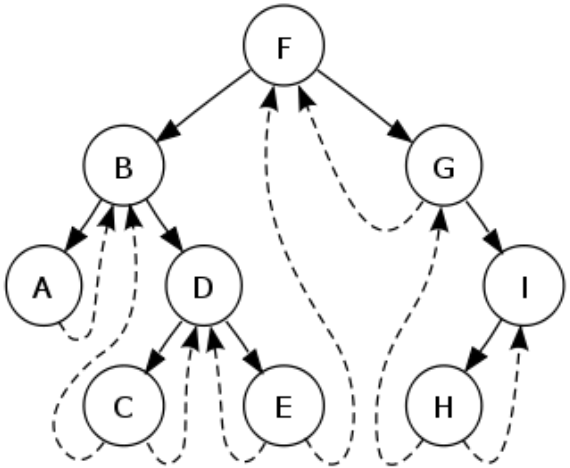
KyuDong SIM

1. 이번 주 실습 내용

- Threaded Binary Tree
  - Left Insertion

Threaded binary tree

- Binary tree 중 한가지
- Leaf node에서 parent로 접근하는 pointer를 가짐



Threaded binary tree

- Structure

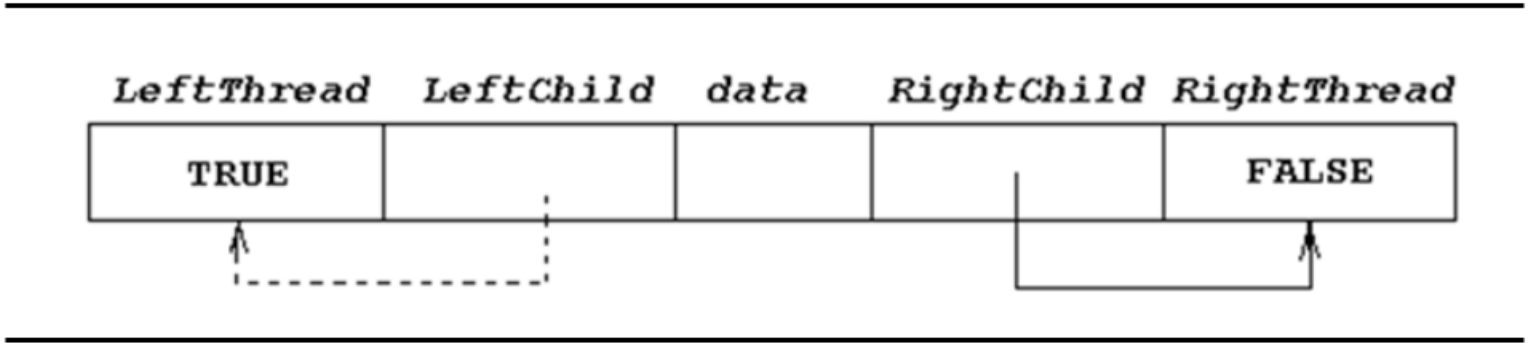
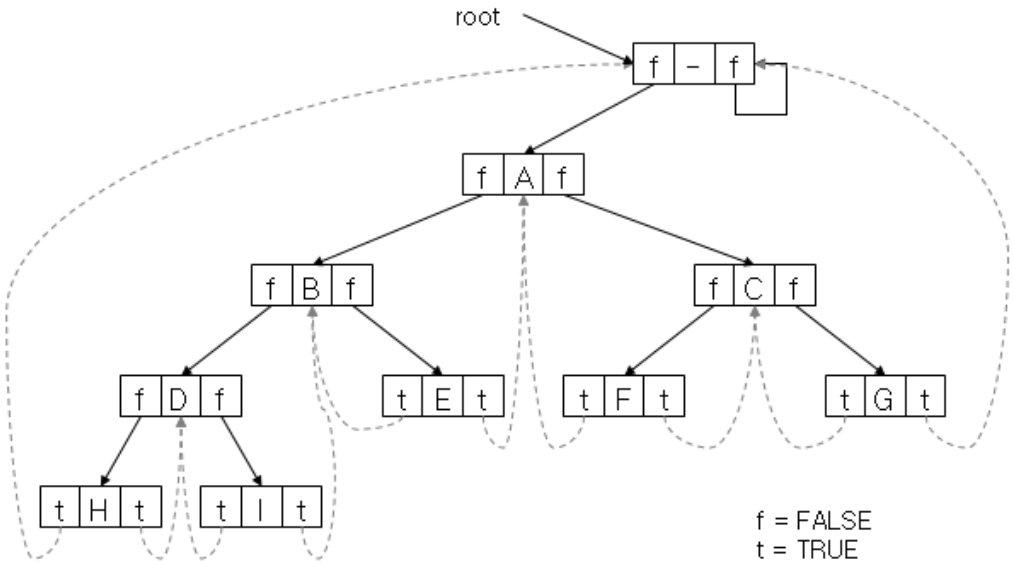


Figure 5.22 : An empty threaded binary tree

Threaded binary tree

- Structure



Right Insertion

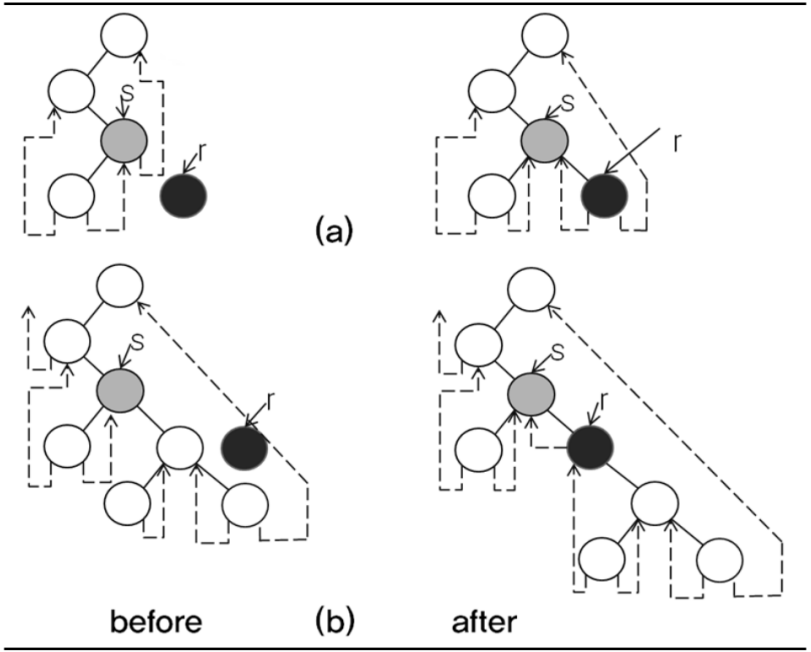
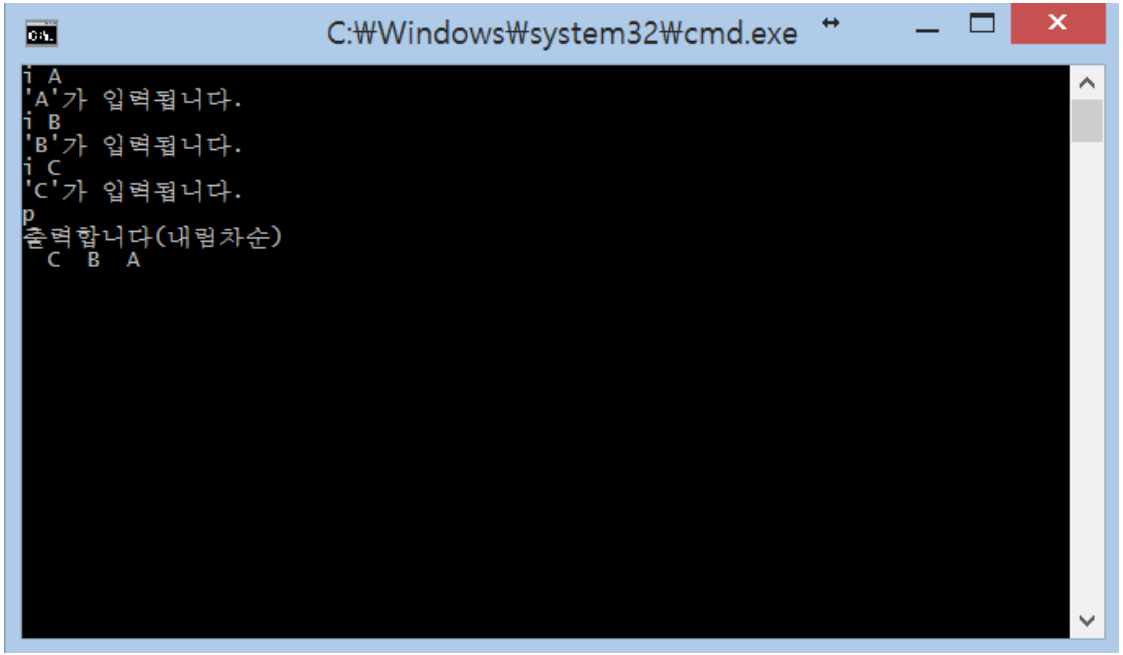


Figure 5.24 : Insertion of r as a right child of s in a threaded binary tree

결과 예 (insertLeft)



Tree에 leftInsert 수행
Code 확인으로 검사
(출력은 자율적으로)

Data type

```
typedef struct threadedTree *threadedPointer;
typedef struct threadedTree {
    short int leftThread;
    threadedPointer leftChild;
    char data;
    threadedPointer rightChild;
    short int rightThread;
};
```

insucc

```
threadedPointer insucc(threadedPointer tree)
{
    threadedPointer temp;
    temp = tree->rightChild;
    if (!tree->rightThread)
        while (!temp->leftThread)
            temp = temp->leftChild;

    return temp;
}
```

rightChild의 left leaf 탐색

tinorder

```
void tinorder (threadedPointer tree)
{/* traverse the threaded binary tree inorder */
  threadedPointer temp = tree;
  for (;;) {
    temp = insucc(temp);
    if (temp == tree) break;
    printf("%3c", temp->data0;
  }
}
```

Program 5.11 : Inorder traversal of a threaded binary tree

Threadedpointer의 Inorder 출력

insertRight

```
void insertRight (threadedPointer s, threadedPointer r)
{/* insert r as the right child of s */
  threadedPointer temp;
  r->rightChild = parent->rightChild;
  r->rightThread = parent->rightThread;
  r->leftChild = parent;
  r->leftThread = TRUE;
  s->rightChild = child;
  s->rightThread = FALSE;
  if (!r->rightThread) {
    temp = insucc(r);
    temp->leftChild = r;
  }
}
```

Program 5.12 : Right insertion in a threaded binary tree

제출 및 알림

수업 중 확인 or 메일제출 (학번 기입)

메일 제출 :

주소 : (89kdsim@naver.com)

기한 : ~2015-05-11