

모든 공학의 개념은 이곳에

이 블로그 [자료구조](#) 카테고리 글

자료구조

[자료구조] 트리(Tree)의 개념 및 : 후위순회, 총별순회

[자료구조] 트리(Tree)의 개념 및 전위순회, 중위순회, ...
2012. 3. 31.



엔지니

2012. 3. 31. 23:35

안녕하세요! ㅋㅋ

자료구조는 아직 포스팅할 예정이 없었는데 매틀랩 자료를 올

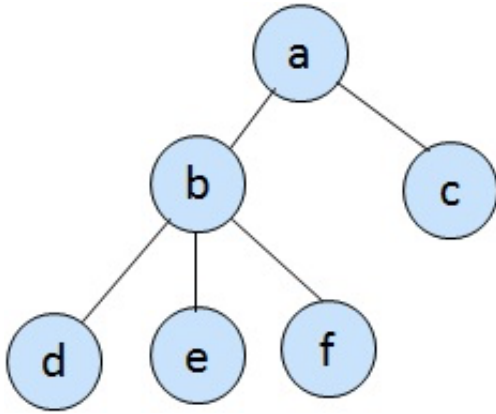
것 같아 이렇게 포스팅하게 됩니다.

어쩌면 이번기회를 통해 c를 이용한 자료구조도 포스팅할지도

그럼 시작하겠습니다!

Tree(나무구조)의 정의

- ◎ 1개 이상의 유한한 개수의 노드(or vertex)의 집합
- ◎ 루트 노드와 0개 이상의 겹치지 않는(disjoint) 하위 나무 ·



이해를 돕기위해 예시로 위 그림을 첨부했습니다.

그래도 약간 난해하실 수 있으니 설명들어갈게요!

트리는 전체적으로 node(or vertex)와 edge(or link)라는 것

위에 동그라미가 바로 node이며 여기에 우리는 특정 정보를 :

edge라 부르며 정보들간의 관계를 나타냅니다.

기본적인 구조는 이렇구요, 이제 트리에 대한 용어 설명에 들

path : edge에 의해 연결된 node들의 집합

root node : 최상위의 node

parent(부모), children(자식), siblin(형제), grandparent(조

: 기준이 되는 것의 바로 직계 상위 node를 부모, 바로 아래층

둔 node들을 형제, 부모의 부모를 조부모, 직계 상위 노드들은

leaf(잎) : 자식이 없는 node

subtree : 큰 tree에 속한 작은 tree

node의 degree : 하위 subtree의 개수

node의 level : root node 부터 최하위 node까지의 중첩되기

휴...용어가 조금 많네요^^;;

대충 훑어 보시면 대부분은 아실 수 있습니다.

혹여 궁금한 것이 있으시다면 댓글을 남겨주시구요

이제 이진나무에 대해 알아보까 합니다.

Binary Tree(이진 나무, 이진 트리)의 정의

모든 내부 node들이 둘 이하의 자식node를 갖는 나무, 노드

root node를 기준으로 왼쪽 이진나무, 오른쪽 이진나무로 이

◎ Complete Binary tree(완전 이진나무)

가장 마지막 level을 제외한 모든 node들이 꽉 차있고 마지막

node까지 빈 칸이 없는 tree

◎ Full Binary Tree(포화 이진나무)

마지막 level까지 완전히 꽉 차있는 이진 트리를 말함

이제 c 코드로 구현한 이진트리를 볼까요?

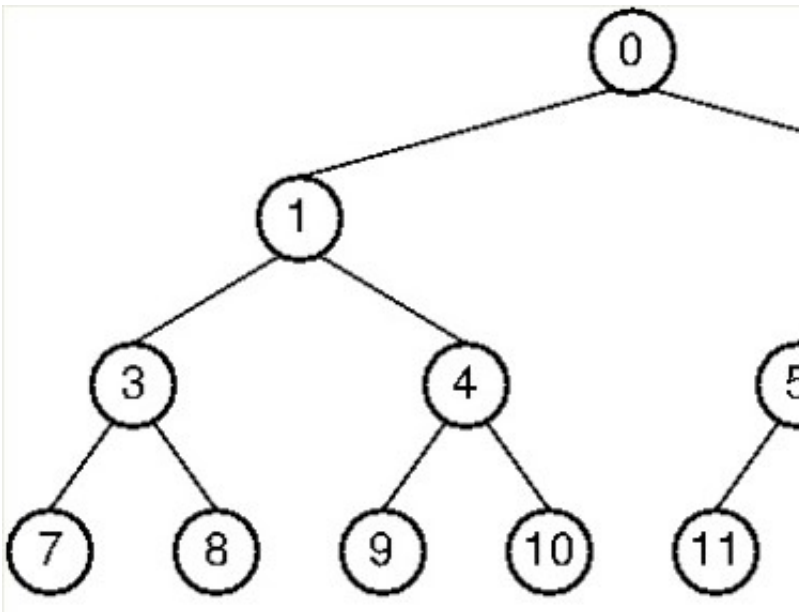
```
typedef struct _node {  
    int key;  
    struct _node *left;  
    struct _node *right;  
} node;  
  
node *head, *tail;
```

위와 같이 구조체를 사용하여 코딩할 수 있습니다 참고하세요

다음은 전위 순회, 중위 순회, 후위 순회, 층별 순회에 대해 알아

이진 나무 순회(Tree Traverse)

- ⊙ 전위 순회(preorder traverse) : 뿌리(root)를 먼저 방문
- ⊙ 중위 순회(inorder traverse) : 왼쪽 하위 트리를 방문 후 !
- ⊙ 후위 순회(postorder traverse) : 하위 트리 모두 방문 후
- ⊙ 층별 순회(level order traverse) : 위 쪽 node들 부터 아래



위와 같은 트리가 있다고 한다면 각각 순회방법은 다음과 같습

전위 순회 : 0->1->3->7->8->4->9->10->2->5->11->6

중위 순회 : 7->3->8->1->9->4->10->0->11->5->2->6

후위 순회 : 7->8->3->9->10->4->1->11->5->6->2->0

층별 순회 : 0->1->2->3->4->5->6->7->8->9->10->11

★전위 순회는 뿌리->왼쪽 자식->오른쪽 자식 순

★중위 순회는 왼쪽자식-> 뿌리-> 오른쪽 자식

★후위 순회는 왼쪽자식->오른쪽 자식-> 뿌리

★층별 순회는 그냥 노드의 순서대로

이 정도 만 아시면 어느정도 순회방식에 대한 개념은 잡혔다고

c코드 구현 방식도 올려드릴테니 참고하시면 좋을 것 같습니다

```
void preorder_traverse(node *t) {
    if (t != tail) {
        visit(t);
        preorder_traverse(t->left);
        preorder_traverse(t->right);
    }
}
```

```
void inorder_traverse(node *t) {
    if (t != tail) {
        inorder_traverse(t->left);
        visit(t);
        inorder_traverse(t->right);
    }
}
```

```
void postorder_traverse(node *t) {
    if (t != tail) {
        postorder_traverse(t->left);
        postorder_traverse(t->right);
        visit(t);
    }
}
```

```
void levelorder_traverse(node *t) {
    put(t);
    while (!is_queue_empty()) {
        t = get();
        visit(t);
        if (t->left != tail) put(t->left);
        if (t->right != tail) put(t->right);
    }
}
```

자료구조에서 트리는 반드시 배우는 것이니 꼭 알아두시면 좋

매틀랩에서의 다음 포스팅에선 트리와 그 순회방법에 대해 코

댓글 부탁드립니다 ^^

#프로그래밍 #자료구조 #트리 #tree #전위순회 #후위순회 #

29

5



엔지니

엔지니의 공학 블로그. 서로이웃 신청 환영해요^^

이웃추가

이 블로그의 **#프로그래밍** 다른 글

[\n 받 2\nc 2\nc

↳

우

20

—

C

20

—

C

20



맨 위로



blog market | 위드폴리
제철 산지의 신선함을 전해요! 과일 마켓

PC버전으로 보기

295