Day 13(20 02 04) \_자료구조

트리 (tree)

-계층적 구조

-비선형 자료구조 (1:N)

Root : 트리의 최 상단 노드

Depth : 루트로부터 어떤 노드 까지의 거리

Level : depth가 2인 노드들의 집합

Leaf(Terminal), 말단 노드 : tree에서 자식 노드가 없는 노드

형제node : 부모 노드가 똑 같은 애들

Height : 말단 노드 중 root로부터 가장 멀리 떨어진 노드의 depth

Degree(차수) : 형제 노드의 수

tree의 차수 : 모든 노드의 차수 중 가장 큰 것 (형제로 되어있어야 한다)

이진트리(binary tree) – 각 노드당 자식 노드가 2개 이하인 것. 말단 노드 제외

포화이진트리 – 말단 노드를 제외한 모든 노드들이 자식 노드가 2개씩 있음

\*이진 탐색 트리의 insert – 루트보다 작으면 왼쪽.. 자식보다 작으면 왼쪽.. 자식보다 크면 오른쪽..

내가 한 insert

|  |
| --- |
| void Insert(Tree\*\* head, int data) {  Tree\* tmp;  if ((\*head) == NULL) {  (\*head) = GetNode();  (\*head)->data = data;  return;  }  else {  if ((\*head)->data > data) {  (\*head)->left = GetNode();  (\*head)->left->data = data;    }  else {  (\*head)->right = GetNode();  (\*head)->right->data = data;    }    }  } |

선생님이 한 insert

|  |
| --- |
| void Insert(Tree\*\* root, int data) {  if ((\*root) == NULL) {  (\*root) = GetNode();  (\*root)->data = data;  return;  }  else if ((\*root)->data > data) {  Insert(&(\*root)->left, data);  }  else {  Insert(&(\*root)->right, data);  }        } |

순회(TRAVERSAL)

A

B c

-Preorder (root – left – right) : A – B-C

-Inorder(left – root – right) : B- A- C

-Postorder(left-right-root) : B-C-A-