

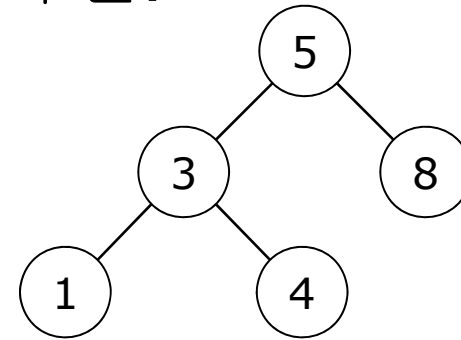
실습 8 – 트리의 기본 연산

- 실습 목표
 - 노드 표현을 이용하여 트리를 구성한다.
 - 트리의 기본 연산(순회, 복사, 동질성 검사) 알고리즘의 동작 과정을 이해한다.

트리 구성 및 순회 알고리즘

- `struct node {int data; struct node *lchild; ...*rchild;}`
- **(8.1.)** main에서 직접 코드로 트리를 구현:

- `A = make_tree_by_code()`
- 다섯 개의 노드를 할당
- 오른쪽 트리를 직접 생성
- 루트 노드를 return



- 3가지 순회 알고리즘(inorder, preorder, postorder)을 구현
 - main에서 각 알고리즘을 호출하고, 실행 결과를 확인

트리의 기본 연산

- **(8.2.)** 트리 복사 알고리즘(프로그램 5.6)을 구현하고 확인
 - $B = \text{copy}(A)$
 - B에 대해서도 순회 알고리즘 호출하고 결과 확인
- 이진 트리의 동일성(프로그램 5.7)을 구현
 - $\text{equal}(A, B)$ 의 결과 출력
- 트리를 복사하되, 왼쪽 자식과 오른쪽 자식을 교환하여 복사하는 **swap** 알고리즘(그림 5.20)을 구현
 - $C = \text{swap}(A)$
 - 순회 알고리즘을 실행하여 결과 확인
 - $\text{equal}(A, C)$ 의 결과 출력

트리의 자동 생성

- **(8.3.)** 정수 n 을 입력받아, 1부터 n 까지 데이터를 저장하는 이진 트리를 생성
 - 루트 노드의 데이터: 1
 - 2는 루트의 lchild 혹은 rchild?
 - 선택 방식: `rand()`의 결과를 2로 나누어 나머지가 0이면 왼쪽, 1이면 오른쪽에 저장
 - 3은 어디에?
 - 선택 방식에 따라 일단 왼쪽 또는 오른쪽을 정하고,
 - 그 방향에 `child`가 있으면 내려가고, 다시 선택
 - `child`가 없으면 그 자리에 추가
 - 이 과정을 n 번 반복
- 생성된 트리에 대해 **inorder**와 **preorder**를 출력한 후, 생성된 트리의 모양을 맞추기