|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 『자료 구조』 실습 워크북 (L6) TREE (2) 이진 탐색 트리 | | | | | |
| 학과 |  | 학번 |  | 이름 |  |

|  |  |
| --- | --- |
| 실습 1 | <플래쉬8.4> 이진 탐색 트리 |
| 실습 목표 | 이진 탐색 트리의 특징을 이해한다. |
| 실습 내용 | <플래쉬8.4>를 살펴본다. (※ L6 실습자료는 L5 실습자료를 이용한다.) |

|  |  |
| --- | --- |
| 실습 2 | <플래쉬8.5><플래쉬8.6> 이진트리의 생성 방법 (삽입, 삭제) |
| 실습 목표 | 이진 트리의 생성 방법 (삽입, 삭제)을 이해한다. |
| 실습 내용 | <플래쉬 8.5>와 <플래쉬 8.6>를 살펴본 후, 다음과 같은 트리에서 17을 삭제했을 경우, 어떻게 변경되는 지를 그려라. |
| 실습 결과 |  |

|  |  |
| --- | --- |
| 실습 3 | <가상실습 8.2 (애플릿 72)> 이진 탐색 트리 (삽입, 삭제) |
| 실습 목표 | 이진 탐색 트리의 삽입/삭제 동작 원리와 코드를 이해한다. |
| 실습 내용 | 1. 연산의 종류를 ‘삽입’으로 선택, 숫자들의 개수를 ‘6’으로 선택하고, { 50, 16, 83, 8, 17, 71 }을 순서대로 삽입하면서, 코드와 동작 원리를 확인하라.    * “다음” 버튼을 눌러서, 코드를 한 줄씩 실행하면서 노드가 삽입되는 과정과 코드를 함께 살펴보고, 6개의 노드가 삽입된 최종 출력 결과를 기술한다.  (※ 가상실습의 코드는 교재와는 달리 iterative version으로 구현되어 있음에 주의할 것. 교재의 recursive version으로 구현된 코드도 살펴볼 것.) 2. 연산의 종류를 “삭제”로 선택하고, 화면의 노드 중 { 16, 83, 71, 17, 50, 8 } 순서로 삭제하면서 코드와 동작 원리를 확인하라.    * “다음” 버튼을 눌러서, 코드를 한 줄씩 실행하면서 노드가 삭제되는 과정과 코드를 함께 살펴보고, 16이 삭제된 후의 결과를 기술한다.  (※ 가상실습의 코드는 교재와는 달리 iterative version으로 구현되어 있음에 주의할 것. 교재의 recursive version으로 구현된 코드도 살펴볼 것.)) |
| 실습 결과 | (1) 트리 생성(삽입) 결과 (50, 16, 83, 8, 17, 71 순서로 삽입 후 최종 결과)  (2) 트리 삭제 결과 (16 삭제 후 트리 모양) |

|  |  |
| --- | --- |
| 실습 4 | <프로그램 8.14> 영어 사전 프로그램 |
| 실습 목표 | 이진 탐색 트리를 활용한 프로그램을 구현한다. |
| 실습 내용 | 1. <프로그램 8.14>과 강의노트의 수정사항을 이용하여 만든 영어 사전 프로그램에 아래의 단어들을 순서대로 삽입하였을 때, 생성되는 이진 탐색 트리를 그려보시오.   (link 연결), (node 노드), (apple사과), (pineapple파인애플), (tree 나무), (love 사랑)   1. 이진 탐색 트리 구성 후, 단어 love와 tree를 찾을 때, 탐색 횟수를 계산하시오. (탐색 횟수는 3-way key comparison 횟수로 한다.) 2. 단어수가 n개인 단어장을 탐색할 때, 시간 성능 (time complexity)를 worst case, average case 에 대해 계산하고, worst case가 어떤 경우인지 설명하시오. |
| 실습 결과 | 1. 이진 탐색 트리 모습 2. 탐색 횟수   Love: 회 / tree: 회   1. Worst case:   Average case:   1. 아주Bb에 파일 L06\_No4\_ BSTDic\_학번\_이름.c 로 제출 |

|  |  |
| --- | --- |
| 회고 | 이진 탐색 트리를 활용하여 해결가능 한 사례를 생각해보자. (예: 영어 사전 기능)  어떤 특징을 갖는 문제가 이진 트리로 해결가능한지 생각해보자.  기타 소감: |