

알고리즘 설계과제 1

[앱스토어 관리 프로그램 설계]

1. 과제 개요

김인하는 레드블랙트리(red-black tree)를 이용하여 앱스토어에 등록된 애플리케이션들을 효율적으로 관리하는 프로그램을 설계하려 한다. 이 프로그램은 판매하는 애플리케이션 관리를 위한 몇 가지 기능들을 수행한다. 각 기능에 대한 정보를 표준입력으로 입력받고 수행한 결과를 표준출력으로 출력하는 프로그램을 구현하시오.

2. 애플리케이션 정보 형식

- ① ID (기준키, 유일함): 1,000,000 ~ 1,999,999 사이의 정수.
- ② 이름: 공백 없는 30 bytes 이내의 문자열. 예) "Notepad"
- ③ 용량: 애플리케이션 저장용량. 1 ~ 10,000 사이의 정수(단위: MB) 예) "303"
- ④ 가격: 애플리케이션 구매 가격. 0 ~ 1,000 사이의 정수(단위: 천원)

각 줄에 ①~④의 정보들이 차례로 공백으로 구분되어 저장된다.

필드 ①	필드 ②	필드 ③	필드 ④
ID	이름	용량	가격

3. 수행할 기능

(1) 애플리케이션 등록

- 질의형식: "I A N S P"

I: 새로운 애플리케이션을 등록하는 질의를 나타내는 기호

A: ID

N: 이름

S: 용량

P: 가격

- 출력형식: "De"

De: 트리에서 애플리케이션 정보가 저장된 노드의 깊이 (root의 깊이는 0)

- 설명: 애플리케이션 정보를 입력받아 레드블랙트리에 노드를 삽입하고, 그 노드의 깊이를 출력한다. 만약 해당 애플리케이션이 이미 등록되어 있다면, 그 노드의 깊이를 출력하고 등록은 거절된다.

(2) 애플리케이션 검색

- 질의형식: "F A"

F: 특정 애플리케이션을 검색하는 질의를 나타내는 기호

A: ID

- 출력형식: "De N S P" 또는 "NULL"

De: 트리에서 애플리케이션 정보가 저장된 노드의 깊이

N: 이름

S: 용량

P: 가격

- 설명: 입력으로 주어진 애플리케이션 ID를 탐색하여 존재하면, 애플리케이션의 정보를 출력한다. 만약 애플리케이션이 존재하지 않으면, "NULL"(주의: 대·소문자 정확히 일치시킬 것)을 출력한다.

(3) 애플리케이션 업데이트

- 질의형식: "R A N S P"

R: 애플리케이션을 업데이트 질의를 나타내는 기호

A: ID

N: 업데이트할 이름

S: 업데이트할 용량

P: 업데이트할 가격

- 출력형식: "De" 또는 "NULL"

De: 트리에서 애플리케이션 정보가 저장된 노드의 깊이

- 설명: 입력으로 주어진 애플리케이션 ID를 탐색하여 존재하면, 입력받은 애플리케이션 정보를 업데이트하고, 애플리케이션에 대한 노드의 깊이를 출력한다. 만약 애플리케이션이 존재하지 않으면, "NULL"을 출력한다.

(4) 애플리케이션 할인

- 질의형식: "D x y P"

D: 애플리케이션을 할인하는 질의를 나타내는 기호

x: 범위의 시작을 나타내는 정수

y: 범위의 끝을 나타내는 정수

P: 할인을

- 출력형식: 없음

- 설명: 범위 [x, y] 내의 ID를 가진 애플리케이션을 모두 탐색하여 각 애플리케이션의 가격에 'P'의 할인을 적용한다.

$$\text{할인가} = \text{가격} \times \frac{100 - P}{100} \quad (\text{단, 할인가의 소수 부분은 버린다.})$$

4. 입출력 제한사항

- (1) 질의는 최대 100,000개가 입력된다.
- (2) 전체 질의에 대해 2초의 제한 시간 이내에 수행되어야 한다.
- (3) 제시한 입출력 형식대로 표준입출력을 사용하여 처리한다.
- (4) 문제에서 설명되지 않은 예외 처리할 질의는 입력되지 않는다.

5. 프로그램 입출력 예

(1) 표준입출력 예시

첫 번째 줄은 질의의 수가 주어지고, 이후 각 줄에는 질의 정보가 주어진다.

파란색: 프로그램 표준입력 내용

빨간색: 프로그램 표준출력 내용

```
=====
20
I 1350226 Internet 1003 12
0
I 1286676 Internet 7439 15
1
I 1510572 InhaUniv 5426 3
1
I 1774327 Viewer 916 30
2
I 1057394 Delivery 6466 1
2
I 1778330 Camera 677 48
2
I 1262909 Notepad 9185 22
1
I 1172552 Word 6928 30
3
I 1164042 Delivery 7501 2
2
I 1346891 Media 5451 956
3
F 1262909
1 Notepad 9185 22
R 1262909 Notepad2 9655 24
1
F 1262909
1 Notepad2 9655 24
R 1165725 Game 9250 33
NULL
I 1165725 Calendar 3221 5
3
R 1165725 Game 9250 33
3
F 1165725
3 Game 9250 33
D 1770000 1780000 10
F 1774327
2 Viewer 916 27
F 1778330
3 Camera 677 43
(프로그램 종료)
```

(2) 표준입출력 예시에서의 각 입/출력 내용 (구현을 검증할 때, 활용할 것)

Input
20 I 1350226 Internet 1003 12 I 1286676 Internet 7439 15 I 1510572 InhaUniv 5426 3 I 1774327 Viewer 916 30 I 1057394 Delivery 6466 1 I 1778330 Camera 677 48 I 1262909 Notepad 9185 22 I 1172552 Word 6928 30 I 1164042 Delivery 7501 2 I 1346891 Media 5451 956 F 1262909 R 1262909 Notepad2 9655 24 F 1262909 R 1165725 Game 9250 33 I 1165725 Calendar 3221 5 R 1165725 Game 9250 33 F 1165725 D 1770000 1780000 10 F 1774327 F 1778330

Output
0 1 1 2 2 2 1 3 2 3 1 Notepad 9185 22 1 1 Notepad2 9655 24 NULL 3 3 3 Game 9250 33 2 Viewer 916 27 3 Camera 677 43

6. 주의 사항 (지키지 않으면, 0점 또는 감점)

(1) 개발언어 및 채점서버 환경

- ① 허용된 개발언어: C, C++
- ② OS: Ubuntu 18.04 (64-bit)
- ③ gcc 버전: gcc (Ubuntu 7.3.0-16ubuntu3) 7.3.0 (c++14 지원)

(2) 제출 파일

- ① 보고서 (I-class에 제출)
 - (a) 파일형식: 아래아한글 문서(.hwp), MS Word 문서(.doc, .docx), PDF 문서(.pdf)
 - (b) 양식: 첨부된 파일 참조
 - (c) 파일명: "학번_이름"으로 통일 (예: 12059876_홍길동)
 - ② 소스코드 (채점서버에 제출)
 - (a) 하나의 소스코드 파일로 구현하여 서버에 제출 및 채점 (예: 12059876_GildongHong.cpp)
(채점서버에서 파일명에 대한 한글 인식이 되지 않기 때문에, 파일명은 영문이름으로 제출)
 - (b) 소스코드에 반드시 주석이 기재돼 있어야 함
- ※ 파일명 형식 어길 시, 감점

(3) 기타

- ① 제출 마감: 11월 14일(일) 23:59까지 (마감 후에 제출되는 과제는 받지 않음)
- ② 마감 직전에는 I-class 및 채점 서버에 학생들이 많이 몰리기 때문에 혼잡할 수 있으니, 마감 시간보다 1시간 정도 이전에 여유 있게 제출하는 것을 권장
- ③ 제출 후에는 제대로 제출되었는지 반드시 확인
- ④ 부정행위가 적발될 경우, 베낀 학생과 원본을 제공한 학생 모두 0점 처리 및 성적의 총점에서 -10점 감점. 인터넷이나 책(교재, 참고서 등)의 소스코드, 예전에 제출했던 과제 등을 베껴서 제출해도 부정행위로 처리