

과제 5. 트리

과제 5

- ◆ 제출 마감일 : 6월 12일 (수요일) 10:59pm
- ◆ eClass 과제방에 제출
- ◆ 사용 언어
 - C / Python 중 자유 사용 (단, 모든 코드는 직접 작성해야 하며, 이미 구현된 라이브러리를 가져오면 안된다)
- ◆ 제출 양식
 - 보고서 (표지 + 문제 내용과 해결 방안 + 결과 캡처)
 - 소스코드 (**별도 파일로 제출**)
- ◆ 보고서는 PDF 로 제출

(1) 사전 탐색 트리 만들기

◆ 제공된 파일을 이용해 탐색 트리(A)를 만든다.

- randdict.txt 는 약 48000여개의 단어가 정렬되지 않은 상태로 저장되어 있다.
- 사전 파일을 이용해 이진 탐색 트리를 만들어라.
 - 트리를 구성할 때에는 단어를 하나씩 가져와 구성하라. 단어는 순차적으로 하나씩 트리의 노드로 추가한다.
 - 탐색을 위한 key는 단어를 이용하라.
- 해당 이진 탐색 트리의 전체 높이를 표시하라.
- 이진 탐색 트리의 탐색을 통해 몇 개의 노드가 있는지 표시하라. 단어의 입력의 수를 세지 말고 트리의 노드 수를 세야 한다.
- 트리의 높이를 최적화하는 노력은 하지 않는다. 파일에 저장된 단어를 순서대로 받아 이진 트리를 만드는 것이다.
- 트리는 직접 만들어야 한다.

자유 언어

(1) 사전 탐색 트리 만들기

◆ 실행 예

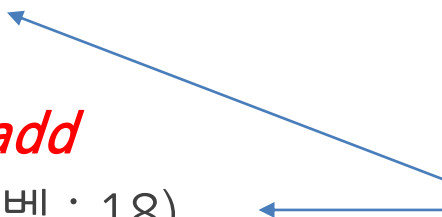
사전 파일을 모두 읽었습니다. 48406개의 단어가 있습니다.
트리의 전체 높이는 42 입니다. 트리의 노드 수는 48406개 입니다.

단어를 입력하세요: *add*

vt. 추가하다 (레벨 : 18)

단어를 입력하세요:

실행 예시이며, 여러분의
결과와 다를 수 있음

Two blue arrows originate from the text '실행 예시이며, 여러분의 결과와 다를 수 있음'. One arrow points to the text '트리의 전체 높이는 42 입니다.' and the other points to the text '트리의 노드 수는 48406개 입니다.'

* 단어의 레벨은 구성된 트리에 따라 다를 수 있다. 단, 전체 높이를 초과하지는 않는다.

(2) 사전 탐색 트리 개선하기

◆ 트리의 높이를 낮추어 보자.

- 트리의 높이가 높으면 자료 검색에 시간이 오래 걸린다.
- 트리의 높이를 낮추는 방법을 생각해보고 적용한다.
- 그 결과를 (1)에서와 같은 결과로 확인한다.
- 우리가 앞에서 배운 내용을 적용한다(힌트 : 정렬).

쉬운 접근법을 찾아서 적용하자.

사전 파일을 모두 읽었습니다. 48406개의 단어가 있습니다.
트리의 전체 높이는 18 입니다. 트리의 노드 수는 48406개 입니다.

단어를 입력하세요: *add*

vt. 추가하다 (레벨 : 13)

단어를 입력하세요:

실행 예시이며, 여러분의
결과와 다를 수 있음

(3) 수식 트리 만들기

◆ 수식 트리를 만든다.

- 수식은 +, -, *, /, () 연산자만 가능 (** 없음)
- 숫자는 실수
- 중위 표기의 수식을 입력하면 수식 트리를 만든다.
- 수식 트리가 제대로 만들어졌다면 순회 순서에 따라 전위, 중위, 후위 표기로 수식을 표현할 수 있다. 레벨 순회도 추가하여 표시한다.
- 수식의 결과값을 화면에 표시하라. 단, 수식 트리에서 결과값을 산출해야 하며, eval() 을 써서는 안 된다.

수식 트리를 반드시 만들어야 한다.
결과만 나오게 만들어서는 안된다.

(3) 수식 트리 만들기

◆ 실행 예

수식 > $3+4*2$

전위표기 : $+ 3 * 4 2$

중위표기 : $3 + 4 * 2$

후위표기 : $3 4 2 * +$

레벨 순회 : $+ 3 * 4 2$

결과값 : 11

(4) 미로 분석기

◆ 주어진 미로를 분석하여 길이 있는지, 여러 개 있는지를 분석하고 미로의 길을 표시한다.

– 과제 1에서 다룬 미로 파일과 같은 유형의 미로 파일을 각자가 하나 만든다.

– 미로 파일을 분석하여

- 길이 있음
- 길이 없음
- 길이 2개 이상 있음

을 분석하여 결과를 표시한다.

– 길이 있는 경우 길을 미로와 함께 표시한다. 길이 여러 개 있을 때에는 그 중 하나만 표시한다.

– 미로의 시작점은 왼쪽 최상단, 미로의 탈출점은 오른쪽 최하단이다.

과제1의 자료구조를 유지할 필요는 없다. 각 과제에 맞는 자료구조가 있다.

(4) 미로 분석기

◆ 실행 예)

– 미로 파일의 이름을 입력하세요 : maze1.txt

분석 결과 : 길이 1개 있음 (길이 없음/길이 2개 이상 있음)

```
+-----+-----+--+
| 0          |          |  | |
| 0+---+ +---+ +  |  |
| 00000|  |          |  |
+---+0|  + +--+  |  |
| 00000|  |          |  |
| 0+--+ +--+--+ +--+ +  |
| 0|  |  |  |  |  |  |
| 0|  +--+  |  +--+ +--+ -|
| 0|  |000|  |          |  |
| 0+  |0+0+---+--+ +  |
| 0  |0|0          |  |  |
| 0+--+0|0+---+---+  |
| 00000|0|0000000  |
+--+---+0|0+---+0+  |
|  |00000|0|00000|  |
|  |0+---+0|0+---+--+
|  |0000000|0000000|
|  +---+ +--+--+ +--+0|
|          |  |  |  |0|
|  +--+  |  +  |  +--+ 0|
|          |  |  |          0|
+--+  |  +  |  + +---+0|
|          |  |          0|
+-----+-----+-----+
```

(5) 지하철 짧은 길 찾기

- ◆ 사용자가 선택한 2개의 역 사이의 짧은 길을 찾아 표시한다.
 - 거리만 고려한다.
- ◆ 서울 시내 지하철 노선도를 보고 40개~50개 지하철 역을 선택하라.
 - 8개 노선에서 역을 고르게 골라야 한다.
 - 환승역을 8개 이상 반드시 포함해야 한다.
 - 자신이 선택한 역으로 구성된 지하철 노선도를 보고서에 넣어라. (테스트를 위해 필요함)
- ◆ 지하철 역 사이의 거리/시간은 다음을 참고하라.
 - <https://data.seoul.go.kr/dataList/OA-12034/S/1/datasetView.do>

(5) 지하철 짧은 길 찾기

◆ 정보 입력은 자신이 직접 정의한 파일을 이용하라.

- 파일은 텍스트 파일을 이용하라(바이너리 파일은 수정이 어려움). 파일의 이름은 subway_자신의학번.txt 로 정하라.
- 자신이 만든 데이터 파일에서 지하철 정보를 가져와 메모리에 지하철 노선도를 구성하라. 따라서 정보를 가지고 있는 데이터 파일을 어떻게 구성할지 설계해야 한다.
- 데이터 파일을 변경하면 프로그램의 실행 결과에 반영될 것이다. 데이터를 소스코드 내에 고정해 두면 안된다.
- 제출된 과제의 테스트는 포함된 데이터 파일을 변경하면서 결과를 확인할 것이다.

(5) 지하철 짧은 길 찾기

- ◆ 출발역과 도착역을 입력하면 최단거리로 얼마만큼 떨어져 있는지 표시하라.
- ◆ 이 과제에서는 환승을 위한 거리/시간은 계산하지 않는다.