

자료구조설계 팀 프로젝트 3주차 회의록

1. 회의 개요

일시	2018.11.18 13시	장소	208관 6층 피시실
작성자	조원희	작성일	2018.11.20
참석자	조원희, 권도경, 박미지, 이용, 김성재 (전원 참석)		
안건	Solving & Experimentation		

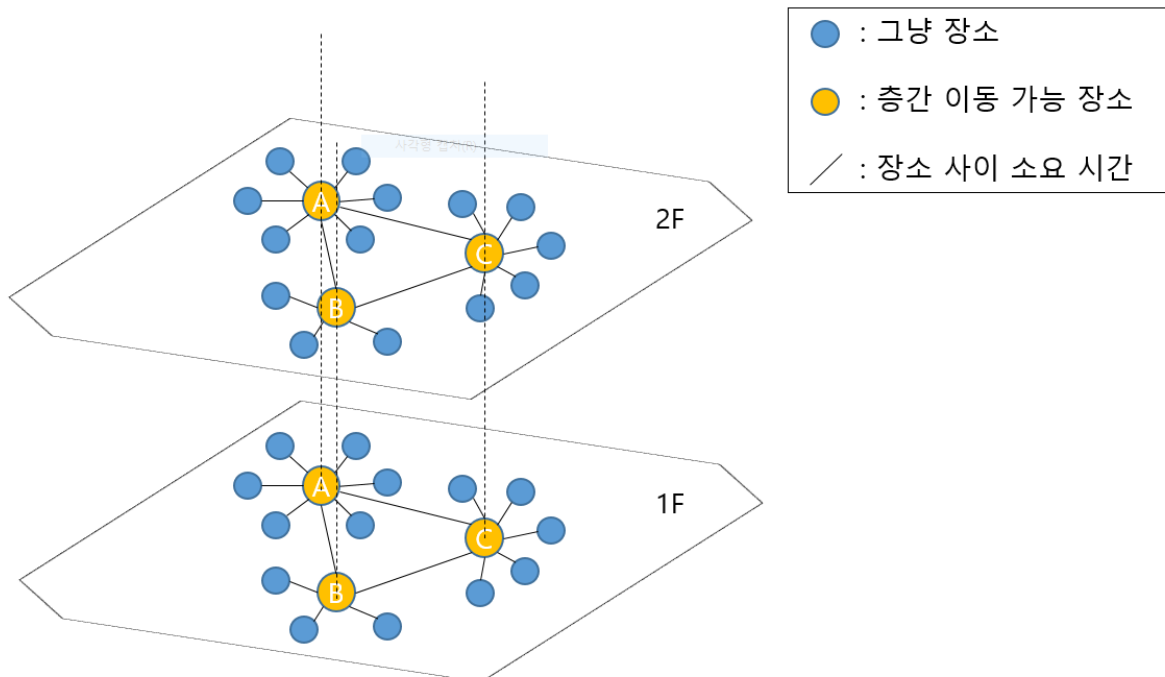
2. 회의 내용

이번 회의에서는 저번 회의에서 결정한 모델링을 약간 수정하고 어떻게 구현을 하고 자료구조를 구성할지에 대한 회의를 진행하였습니다.

우선 단면도를 이용한 310관 graph 모델링 부분에서 약간의 수정사항이 있었습니다.

기존에 같은 층 내부에서의 장소 이동의 경우 모든 장소(Node)를 기준으로 인접 장소(Adjacent Node)와 소요 시간(Weight)을 측정하는 방식을 선택했지만 노드의 수가 생각보다 많아 모든 노드를 기준으로 인접 장소와 소요 시간을 측정하기에는 무리가 있다고 판단했습니다.

그래서 층간 이동이 가능한 장소(Main node)를 크게 A, B, C, D, E 구역으로 지정하고 해당 지점의 인접 장소(Adjacent Node)는 층간 이동이 불가능한 일반 강의실 같은 장소로 지정을 하여 소요 시간을 쉽게 측정하는 방식으로 변경했습니다. 이 방식은 소요 시간을 쉽게 계산할 수 있을 뿐만 아니라 경로 계산시에 경유해야 하는 노드의 수가 비약적으로 줄어 계산 속도가 향상됩니다. 설명은 아래 사진과 같습니다.



자료구조는 우선 강의실을 Node로 두어 Lecture 클래스를 생성했습니다. 이 클래스 안에는 강의실 호수(ex 726) 정보를 갖고 있는 ClassName 변수, 요일별 시간대별 수강인원 정보를 갖고 있는 Monday[] ~ Friday[] 배열 그리고 해당 강의실의 구역(위에서 언급한 A, B, C, D, E 구역 중 하나) 정보를 갖고 있는 nearbyElevator라는 변수를 갖고 있습니다. 즉 Lecture 클래스는 강의실 이름, 요일별 시간대별 수강인원 그리고 층 내에 구역 정보를 갖고 있는 셈입니다. 실제 구현은 아래와 같습니다.

Lecture
<u>ClassName</u> : 강의실 호수
Monday[] : 월요일의 시간대별 수강인원
Tuesday[] : 화요일의 시간대별 수강인원
Wednesday[] : 수요일의 시간대별 수강인원
Thursday[] : 목요일의 시간대별 수강인원
Friday[] : 금요일의 시간대별 수강인원
<u>nearbyElevator</u> : 근처 엘리베이터 구역 이름

```
public class node {
    String ClassName;
    String[] monday = new String[18];
    String[] tuesday = new String[18];
    String[] wednesday = new String[18];
    String[] thursday = new String[18];
    String[] friday = new String[18];
    String nearbyElevator;
}
```

- 강의실의 데이터가 89개 존재
- Lecture의 배열이 89개 생성

- 강의명과 수강인원을 같이 기입했기 때문에 9교시 * 2 인 18의 크기의 배열

엘리베이터의 정보를 갖고 있는 Elevator 클래스도 생성했습니다. 310관에 존재하는 엘리베이터가 12개인만큼 이 클래스 안에는 각 엘리베이터의 운행층을 나타내는 12개의 boolean 배열이 존재합니다. 해당 인덱스의 층이 운행하는 층이면 true값을 아니면 false값을 갖고 있습니다.

Ex) 만약 운행하는 층인 B6를 배열의 '0' 인덱스라고 생각하면 엘베 [0] = true ; 와 같이 저장됩니다.

마지막으로 층 내부에서 이동하는 경우 우리가 모델링한 그래프의 해당 층을 나타내는 자료구조를 생성했습니다. 자료구조는 이중 HashMap으로 구성되었고 첫번째 HashMap에서는 구역(A, B, C, D, E) 이름을 string key로 갖고 있고 그 value 안에 있는 두번째 HashMap에서는 강의실 같은 장소 이름을 string key로 갖고 있고 value는 구역에서 해당 장소까지의 소요 시간인 Edge(Weight)값이 있습니다. 실제 구현은 아래와 같습니다.

HashMap1<"구역 이름", HashMap2<"인접 장소", 소요 시간>>

아직 고려하지 못한 여러 클래스는 다음 회의에서 구현할 예정입니다. 아마 다음 회의에서 프로그램의 전체적인 구성이 정리될 것 같습니다. 추후에 구현을 하면서 더 많고 다양한 자료구조를 사용할 예정입니다.

마지막으로 거의 모든 팀들이 엘리베이터 대기 시간을 계산하는데 많은 어려움을 겪고 있습니다. 우리 팀 또한 많은 고뇌 끝에 자체적으로 '시간대별 엘리베이터 대기 및 이동 시간 공식'을 만들었습니다. 공식은 합리적으로 이벤트를 나누어 그에 따른 확률을 적용하여 만들었습니다.

공식에 대한 설명은 '시간대별 엘리베이터 대기 및 이동 시간 공식.jpg' 파일로 따로 설명했습니다.

<다음 회의까지 할 일>

권도경	데이터 정리, 다음주 발표(11//19) 자료 만들기
김성재	데이터 정리, 구현
박미지	데이터 정리, UI 만들기
이용	데이터 정리, 구현, 다음주 발표(11//19) 준비
조원희	데이터 정리, 구현, 회의록 작성

<회의 사진>

