

안드로이드 스마트폰 상에서의 캠퍼스 네비게이션 시스템의 설계 및 구현

최영웅, 이장호
홍익대학교 컴퓨터공학과
e-mail:gonet0079@gmail.com, janghol@cs.hongik.ac.kr

Design and Implementation of Campus Navigation System on Android Smartphone

Yeongung Choi, Jang Ho Lee
Dept of Computer Engineering, Hongik University
e-mail:gonet0079@gmail.com, janghol@cs.hongik.ac.kr

요 약

작은 캠퍼스를 지닌 대학교는 많은 건물이 거미줄처럼 복잡하게 연결되어 있는 경우가 많아서 신입생들이나 방문객들이 캠퍼스에서 길을 잃고 헤매는 경우가 빈번하다. 이에 본 논문에서는 사용자가 가고 싶은 건물과 층수를 입력하면 현재 위치에서 목적지까지의 최단 경로를 제공하는 안드로이드 스마트폰 상에서의 캠퍼스 네비게이션 시스템의 설계 및 구현을 제시한다.

1. 서론

일반적으로, 대학교 내부의 구조는 매우 복잡하여 익숙하지 않은 사람들은 길을 찾기 어렵기 때문에, 원하는 목적지까지의 최단 경로를 안내해 주는 모바일 네비게이션 시스템의 필요성이 증가하고 있다.

다익스트라 알고리즘(Dijkstra Algorithm)[1]은 가장 대표적이고 효율적인 최단 경로 탐색 알고리즘 중의 하나이다. 본 논문에서는 다익스트라 알고리즘을 사용하여 최단 경로 안내 시스템을 설계하였고, Android API[2], Google Maps API[3], Java 언어를 이용하여 구현하였다.

2. 시스템의 주요 요구사항

LTE 데이터, Wi-Fi 등의 사용이 불가능한 경우를 감안하여, 본 시스템은 서버-클라이언트의 관계를 가지는 복합 시스템이 아니라, 모바일 기기 내에서만 구현되는 단일 로컬 시스템 형태로 구상하였다. 내부 파일에 캠퍼스 내부 구조를 서술해 두고, 구동과 동시에 자료구조로 입력받은 다음, 다익스트라 알고리즘을 사용함으로써 클라이언트-서버 구조를 취하지 않고도 동작할 수 있다.

본 시스템에서 도출되는 결과의 가장 중요한 요구사항은, 안내의 결과로 나오는 경로가 반드시 최단 경로임이 확실해야 한다는 것이다. 만일 확실성이 보장되지 않는다면 시스템의 신뢰성이 떨어지며, 제대로 된 결과를 구현했다고 할 수 없다. 다익스트라 알고리즘은 최적의 결과를 반드시 보장하므로, 자료를 실제와 다르게 입력하거나 실제 구현 과정에서 착오를 하지 않는 이상 옳은 결과가 보증된다.

입력 형태는 출발지와 목적지를 건물 이름+층수로 명시하는 형태로 한다. 현 위치를 자동으로 파악하여 실행한다면 매우 편리하지만, 좌표로는 수직(현재 위치하는 층수)을 판정할 수 없기 때문에, GPS를 통해 건물 위치보다 자세한(층수, 강의실 등의) 정보를 얻어 경로를 안내하는 것은 불가능하다. 만일 GPS를 이용한 좌표 안내를 하게 된다면 건물과 건물 사이의 안내가 한계이다. 학교 구조 특성상 1층의 출입구 이외의 불규칙한 연결 통로가 존재하기 때문에 건물의 위치 정보만으로는 실용적인 최단 경로를 안내할 수 없게 된다.

위치의 최소 단위를 각 강의실 단위로 정의하지 않은 이유는 강의실들의 위치, 이름 등을 유지·관리해야 할 자료구조의 크기가 매우 커지기 때문이다.

탐색의 결과는 구현이 완전하게 완료된 것이 검증되면, 구글 지도상에 마커와 간단한 정보가 같이 나오는 형태로 나타내어 직관적으로 정보를 전달하는 것으로 계획하였다.

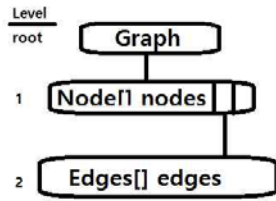
3. 시스템의 자료구조

(그림 1)은 시스템에서 사용하는 자료구조의 계층 구조를 보여주고 있다. Node 클래스는 건물의 각 층에 대한 정보를 가진다. Graph 클래스는 가중치 그래프(Weighted Graph)를 표현하기 위해 모든 Node 갯수만큼의 길이를 가지는 Node형 1차원 배열과 findShortestPath() 등의 실행 메소드를 보유한다.

배열의 각 Node는 자신이 갈 수 있는 Edge 클래스 객체들을 보관하고 있다. Edge 클래스는 출발지, 목적지, 가

중치, 위치 정보 등을 보유한다.

자료 수집을 위해 학교 안을 직접 순회하여 구조를 파악하고 통로를 오가는 데 걸리는 시간을 측정, 기록하였다.



(그림 1) 계층 구조

4. 시스템의 실행 순서

구동과 동시에 지정한 형식을 갖춘 파일을 읽어 Node, Edge 정보를 로드하여 자료구조에 채워 넣는다. 사용자가 메뉴에서 '최단 경로 탐색'을 선택하면, 출발지와 목적지를 대화 상자를 통해 입력받고, 다익스트라 알고리즘을 통해 결과를 도출하여, 올바른 결과가 나온다면 출력한다.

5. 시스템의 사용 예

아래의 결과는 테스트를 위해 본교인 홍익대학교 T동 5층에서 A동 1층으로 가는 길을 본 시스템으로 검색한 결과이다.



(그림 2) 출력 결과



(그림 3) 결과의 해석

구현이 아직 완료되지 않은 관계로 현재 결과는 지도상에 경로로 표시되지 못하고 (그림 2)와 같이 텍스트로 출력되며(화상 키보드 상의 t와 T 사이에 Toast 메시지로 출력), (그림 2)에서 출력되는 번호는 출발지, 도착지를 포함하여 거쳐가야 할 노드의 번호를 역순으로 출력한 것이다. 이 노드를 실제 지도상에 나타내 보면 (그림 3)과 같다. (그림 3)에서 5번 노드는 출발지는 T동 5층을 나타내며, 164번 노드는 지나가야 할 중간 지점을 나타낸다. 30번 노드는 도착지인 A동 1층을 나타낸다.

6. 결론

본 논문에서 제시한 설계 사항을 따라서 구현한 결과, 직접 조사한 자료에 다익스트라 알고리즘을 사용하여 최단 경로를 찾아내는 부분까지는 구현 완료되었다. 이것으로써 복잡한 학교 구조를 자료구조로 표현하여, 자동으로 최단 경로를 찾아내어 편의를 도모한다는 기본적인 설계·제작 의도의 달성에는 성공하였다.

그렇지만 아직 이 결과를 사용자 지정 마커와 선으로 나타내는 그래픽화를 하지 못하여 제작자 이외의 사람이 결과를 읽을 수 없는 상태이다. 차후 일차적인 목표는 결과를 그래픽화로 나타내는 작업을 Google Maps API에서 제공하는 기능을 사용해서 완료, 가독성을 높이는 것이다.

또한 그래픽 작업까지 완료되면 알고리즘 최적화가 차후의 과제이다. 현재 남은 최적화 요소의 대표적인 예시로, 다익스트라 알고리즘에서 다음 방문할 노드를 결정하는 과정에서, 전체 배열 순회 검색이 아니라 거리를 기준으로 한 자바의 우선순위 큐(java.util.PriorityQueue(E))[4]를 미방문 노드 저장용 자료구조로 사용하면 검색 시간이 매우 절약되지만, 새로운 거리를 설정한 이후 힙의 우선순위 갱신이 제대로 되지 않아 원하는 노드를 인출하지 못하는 문제가 발견되어 아직 우선순위 큐를 사용하지 못하고 있다. 이 문제를 수정하여 우선순위 큐를 사용할 수 있도록 수정하는 것을 대표적인 최적화 과제로 들 수 있다.

참고문헌

[1] Richard Neapolitan et al., "Foundations of Algorithms" 4th ed., Jones and Bartlett Publishers
 [2] Android Developers, <https://developer.android.com/index.html?hl=ko>
 [3] Google Maps API, <https://developers.google.com/maps/?hl=ko>
 [4] java.util.PriorityQueue(E), <https://docs.oracle.com/javase/7/docs/api/java/util/PriorityQueue.html>