

HW13 과제: Shortest Path

제출 데드라인: 과제 부여 후 2주 후 24시 까지

예) 금요일 실습일 경우 2주 후 목요일 23:59:59

제출: 클래스룸

질문

pemta81718@gmail.com

검색으로 알 수 있는 간단한 내용은 답변하지 않습니다.

제출파일

subway.h subway.cpp hw13.cpp 학번.tex 학번.pdf
+ 보고서에 첨부한 이미지 파일들

1. 문제

- 준희는 현수네 집에 지하철을 타고 가기로 했다. 이때 최단경로 및 걸리는 시간을 구하여라.
- 현수가 생각해보니 본인의 집이 더럽다며 그냥 밖에서 만나자고 했다. 이때 각자 걸리는 시간의 차이가 최소가 되는 역과 시간을 구하여라.

2. 공지사항

- 소스파일(헤더파일 포함), 결과보고서를 함께 클래스룸에 제출합니다.
- 결과보고서는 치팅 여부를 확인할 수 있는 수단으로 LaTeX로 최대한 자세히 작성합니다.
- C++로 작성합니다. (class 이용)
- 최단경로 알고리즘은 어떤 알고리즘을 참고해도 무방합니다. 단, 사용한 알고리즘을 보고서에 기술합니다.

3. Graph (지하철 노선)

주어진 지하철 노선도는 N개의 노선(line)으로 이루어져 있다. 이동은 각 노선 간에서만 순차적인 방식으로 이동이 가능하며 서로 다른 노선(line)간의 이동은 환승역을 통해서만 가능하다. 같은 호선에서 역 사이 소요시간은 1분이다. 환승에 소요되는 시간은 30초이다. 또한 모든 역은 하나의 노드로 간주된다. (즉, 시청역의 경우 2호선 시청역, 1호선 시청역 이 따로 있는 것이 아니라, 하나의 시청역으로 취급한다.)

ex)

```
City_Hall (line1) -> Seoul_Station (line1) -> Namyeong(line1) = 2분
Chungjeongno (line2) -> City_Hall (line2) -> City_Hall (line1) ->
Seoul_Station (line1) = 2.5분
```

4. 요구사항

a. 프로그램이 시작되면 그래프를 구성 한다. 시작 시 노드(노선정보)의 정보를 읽어 들인 다음 그래프(노선)를 구축한다. ex) `./hw11 stations1.txt input1.txt`

b. `stations1.txt`

역(노드)간의 연결정보가 들어있다. `stations` 는 첫 줄에는 연결되는 노드의 수, 그 다음부터 노드들의 연결정보가 주어진다. 각 노드의 숫자는 해당역의 호선과 역 이름으로 구성되어 있다.

```
cat stations1.txt
72
1 Jongno_5(o)-ga 1 Jongno_3(sam)-ga
1 Jongno_3(sam)-ga 1 Jonggak
...
3 Anguk 3 Jongno_3(sam)-ga
1 Jongno_3(sam)-ga 3 Jongno_3(sam)-ga
3 Jongno_3(sam)-ga 3 Euljiro_3(sam)-ga
...
3 Seoul_Nat'l_ Univ._of_Education 3 Nambu_Bus_Terminal
```

c. `input.txt`

`input` 은 출발역의 호선과 역 이름, 도착역의 호선과 역이름이 저장되어 있다.

```
cat input1.txt
3 Anguk
2 Chungjeongno
```

d. 실행결과 예시

최단경로 과정, 소요시간, 중점, 걸린 시간을 각각 출력한다. // 뒤의 내용은 각 출력의 이해를 돕기 위한 요소로, 실제 출력에서는 없음.

```
./hw13 stations1.txt input1.txt
Anguk //출발역
Jongno_3(sam)-ga
Jonggak
City Hall
Chungjeongno //도착역
5:00 //걸린시간
Jonggak //중 점
2:30 //걸리는 시간(크거나 같은 값)
2:30 //걸리는 시간(작거나 같은 값)
```

e. VSCode 또는 makefile 로 컴파일 후 실행

5. hw13.cpp

```
#include "subway.h"
#include <fstream>
int main(int argc, char *argv[])
{
    int numLine;
    int line1, line2;
    string src, dst;
    if(argc != 3) {
        cerr << "Argument Count is " << argc << endl << "Argument must be " << argc << endl;
        return 1;
    }
    fstream fin(argv[1]);
    if(!fin) {
        cerr << argv[1] << " open failed" << endl;
        return 1;
    }
    fin >> numLine;
    //데이터 받는 과정
    //...
    fin.close();
    fstream fin2(argv[2]);
    if(!fin2) {
        cerr << argv[2] << " open failed" << endl;
        return 1;
    }
    fin2 >> line1 >> src;
    fin2 >> line2 >> dst;
    fin2.close();
    //결과 출력
    return 0;
}
```

6. subway.h, subway.cpp // 자유롭게 구현

7. 결과 보고서 내용

- a. 알고리즘 선택 이유
- b. class 설계 내용 및 이유
- c. 그래프 구성 방식
- d. 결과 값 결과 분석
- e. 어려웠던 부분
- f. 제출한 프로그램이 컴파일이나 실행이 안된 경우, 어디까지 구현했는지 혹은 원인

8. 테스트케이스 주의사항

- a. 호선의 개수는 1 이상 9 이하의 자연수이다.
- b. 한 호선 당 최대 100 개의 역을 가질 수 있다.
- c. 역 이름에서의 띄어쓰기가 존재하지 않는다.
- d. 실제 2 호선처럼 순회하는 호선이 있을 수 있다.
- e. 실제 1 호선처럼 나뉘는 호선이 있을 수 있다.
- f. 모든 역은 양방향으로 오고 갈 수 있다.
- g. 한글역은 존재하지 않는다.
- h. 역의 이름은 최대 30 글자이다.

9. 주의사항

- a. header library 는 stack, queue, vector, fstream, string, math, iostream, algorithm, map 및 지금까지 배운 것으로 제한한다.
- b. 코드는 얼마든지 수정 가능하다.
- c. 파일이름을 정확히 해야 한다.
- d. local 변수를 통해 stack 영역을 너무 많이 사용하면 오류가 날 수도 있으므로, 포인터를 통해 heap 영역을 사용하는 것을 추천한다.
- e. diff 명령어를 쓸 예정이므로 cout 이 정확해야 한다.
- f. Cheating 은 F.
- g. 기본적으로 메일로 제출된 과제는 읽지 않는다.

- h. 제출 마감시간 기준으로 4 시간 전부터 마감까지 제출이 불가능하다면, 이메일 제출도 받는다.
- i. 하지만, 제출 마감시간 15 분 전부터 마감시간까지 제출이 가능했다면, 이메일로 제출된 것은 읽지 않는다.
- j. 지각 제출은 마감일로부터 일주일까지 허용, 50% 점수 반영, 조교 메일로 제출