

## HW10 과제: Shortest Path

제출 데드라인: 과제 부여 후 2주 후 24시 까지

목요일 분반: 12월 1일 목요일 24시 까지 (12월 2일 금요일 0시 까지)

금요일 분반: 12월 2일 금요일 24시 까지 (12월 3일 토요일 0시 까지)

제출 명령어: `submit pem_ta hw10x (x=a,b,c)`

제출 확인: `submit pem_ta hw10x -l` (마이너스 엘)

질문

`pemta81718@gmail.com`

제출파일

`subway.h`   `subway.cpp`   `hw10.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 리눅스에 `submit`
- 결과보고서는 치팅 여부를 확인할 수 있는 수단. Latex로 최대한 자세히 작성
- C++로 작성(class 이용)
- 최단경로 알고리즘은 어떤 알고리즘을 참고해도 무방(보고서에 기술)

### 3. Graph (지하철 노선)

주어진 지하철 노선도는 N개의 노선(line)으로 이루어져 있다. 이동은 각 노선간만에만 순차적인 방식으로 이동이 가능하며 서로 다른 노선(line)간의 이동은 환승역을 통해서만 가능하다. 같은 호선에서 역 사이 소요시간은 1분이다. 환승은 30초이다.

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) ./hw10 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

실행결과 첫 줄과, 두번째 줄에는 각각 출발역과 도착역이 예시와 같이 나온 뒤, 그 아래 과정이 나오면 된다. 마지막으로 소요된 시간을 출력해주면 된다.

```
./hw10 stations1.txt input1.txt
```

```
Anguk //출발역
```

```
Jongno_3(sam)-ga
```

```
Jonggak
```

```
City Hall
```

```
Chungjeongno //도착역
```

```
-3/6-
```

```
5:00 //걸린시간
```

```
Jonggak //중 점
```

```
2:30 //걸리는 시간(크거나 같은 값)
```

```
2:30 //걸리는 시간(작거나 같은 값)
```

e. makefile

명령어: cat makefile / vi makefile

```
TARGET = hw10
```

```
OBJS = hw10.o subway.o
```

```
$(TARGET): (tab) $(OBJS)
```

```
(tab) g++ -o $(TARGET) $(OBJS)
```

```
$(OBJS): (tab) subway.h
```

```
clean:
```

```
(tab) rm -f $(OBJS) $(TARGET)
```

## 5. hw10.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. make 명령어를 통해 실행할 예정이므로, 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 정확 해야한다. main 함수에서 cout 은 안 바꾸는 것이 좋다.
- f. Cheating 은 F.
- g. 리눅스가 12 시간 이상 오류 나지 않는다면 기간 연장은 없다.
- h. 기본적으로 메일로 제출된 과제는 읽지 않는다.

- i. 제출 마감시간 기준으로 4 시간 전부터 마감까지 제출이 불가능하다면, 이메일로 제출도 받는다.
- j. 하지만, 제출 마감시간 15 분 전부터 마감시간까지 제출이 가능했다면, 이메일로 제출된 것은 읽지 않는다.