

2020 DS 실습 10

start date = 11월 13일(금)

due date = 11월 27일(금) 23시 59분 59초 (※ 연장불가함)

Submit 방법 : *submit pem_da hw10*분반

1. 문제

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

2. 공지사항

- 소스파일(헤더파일 포함), 결과보고서를 함께 리눅스에 submit
- 결과보고서는 Cheating 여부를 확인할 수 있는 수단으로서 Latex로 최대한 자세히 작성
- C++로 작성(class이용)
- 최단경로 알고리즘은 어떤 알고리즘을 참고해도 무방. (보고서에 기술)

3. Graph (지하철 노선)

주어진 지하철 노선도는 N개의 노선(line)으로 이루어져 있습니다. 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

stations는 첫 줄에는 연결되는 노드의 수, 그 다음부터 노드들이 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. input1.txt

input은 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
```

```

5:00          //걸린시간
Jonggak       //중 점
2:30          //걸리는 시간(크거나 같은 값)
2:30          //걸리는 시간(작거나 같은 값)

```

e. makefile

```

cat makefile

TARGET = hw10

OBJS = hw10.o subway.o

$(TARGET) : $(OBJS)
    g++ -o $(TARGET) $(OBJS)

$(OBJS) : subway.h

clean:
    rm -f $(OBJS) $(TARGET)

```

5. 코드

```

#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. 결과 보고서 내용

- a. 알고리즘의 선택이유
- b. class설계 내용 및 이유
- c. 그래프 구성 방식
- d. 결과 값 결과 분석
- e. 어려웠던 부분
- f. 제출한 프로그램이 컴파일이나 실행이 안 될 경우, 어디까지 구현했는지 혹은 원인

7. 제출파일

- a. subway.h
- b. subway.cpp
- c. hw10.cpp
- d. 학번.tex
- e. 학번.pdf

8. 테스트케이스 주의사항

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

9. 주의사항

- a. header library는 stack, queue, vector, fstream, string, math, iostream, algorithm, map 및 과제 8까지 배운 것으로 제한한다.
- b. 코드는 얼마든지 수정가능하다.
- c. make 명령어를 통해 실행할 예정이므로, 파일이름을 정확히 해야한다.
- d. local 변수를 통해 stack영역을 너무 많이 사용하면 오류가 날 수도 있으므로, 포인터를 통해 heap영역을 사용하는 것을 추천한다.
- e. diff명령어를 쓸 예정이므로 cout이 정확해야한다. main함수에서 cout은 안 바꾸는 것이 좋다.
- f. Cheating은 F.
- g. 리눅스가 12시간 이상 오류나지 않는다면 기간 연장은 없다.
- h. 기본적으로 메일로 제출된 과제는 읽지 않는다.
- i. 제출 마감시간 기준으로 4시간 전부터 마감까지 제출이 불가능하다면, 이메일로 제출도 받는다.
- j. 하지만, 제출 마감시간 15분 전부터 마감시간까지 제출이 가능했다면, 이메일로 제출된 것은 읽지 않는다.

10.질문

- a. pemta806@gmail.com으로 질문한다.
- b. 영상통화 질문을 하고 싶다면 미리 메일로 스케줄을 잡는 것을 추천한다.
- c. 과제 특성상 받을 수 있는 질문의 종류가 제한 될 수 있다.