

# Problem Solving

현재 위치와 이동시간을 고려한 최적의  
영화관 추천

컴퓨터공학부  
20120596  
발표자 이용

# 지하철 노선

수도권 내 모든 지하철의 정보를 수집	1 2 3 소요산 동두천 2 3 3 동두천 보산 3 4 3 보산 동두천중앙 4 5 3 동두천중앙 지행 5 6 3 지행 덕절 6 7 3 덕절 덕계 7 8 3 덕계 주내 8 9 3 주내 녹양 9 10 3 녹양 가평
✓ 각각의 역을 상수로 표현	10 11 3 가평 의정부 11 12 2 의정부 회룡 12 13 3 회룡 망월사 13 14 2 망월사 도봉산 14 15 3 도봉산 도봉 15 16 2 도봉 방학 16 17 2 방학 창동 17 18 3 창동 녹천 18 19 2 녹천 월계 19 20 2 월계 성북 20 21 2 성북 석계 21 22 2 석계 신이문 22 23 2 신이문 외대앞 23 24 3 외대앞 회기 24 25 2 회기 청량리 25 26 2 청량리 제기동 26 27 1 제기동 신설동
✓ 각각의 역에서 역으로 이동시간을 상수로 표현	
✓ 환승 시간을 포함한 최단 경로 고려, 가장 빠른 시간을 Return	

# 지하철 노선도

✓ 인접행렬로 데이터를 구성

✓ 자기 자신의 앞, 뒤 역의 COST를 갖는다

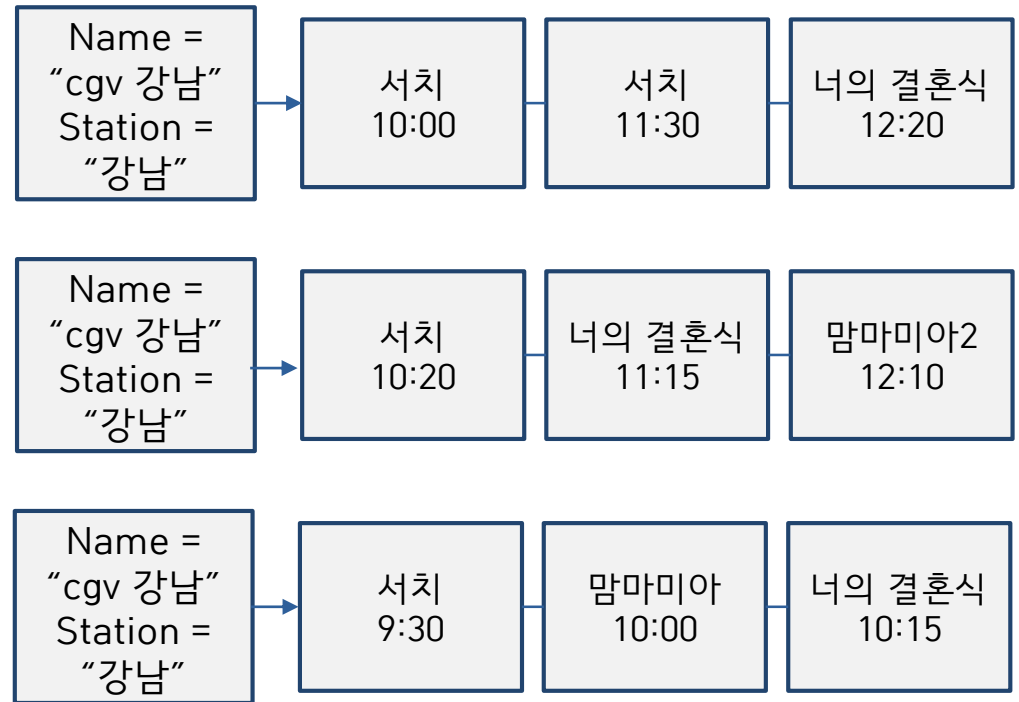
✓ 다익스트라 알고리즘을 통해 모든 역의 소요 COST를 얻는다.

	동두천	소요산	보산
동두천	0	2	123456
소요산	2	0	2
보산	123456	2	0

# 영화관 데이터

- ✓ 영화의 제목은 벡터를 사용하여 배열의 형태로 저장합니다.

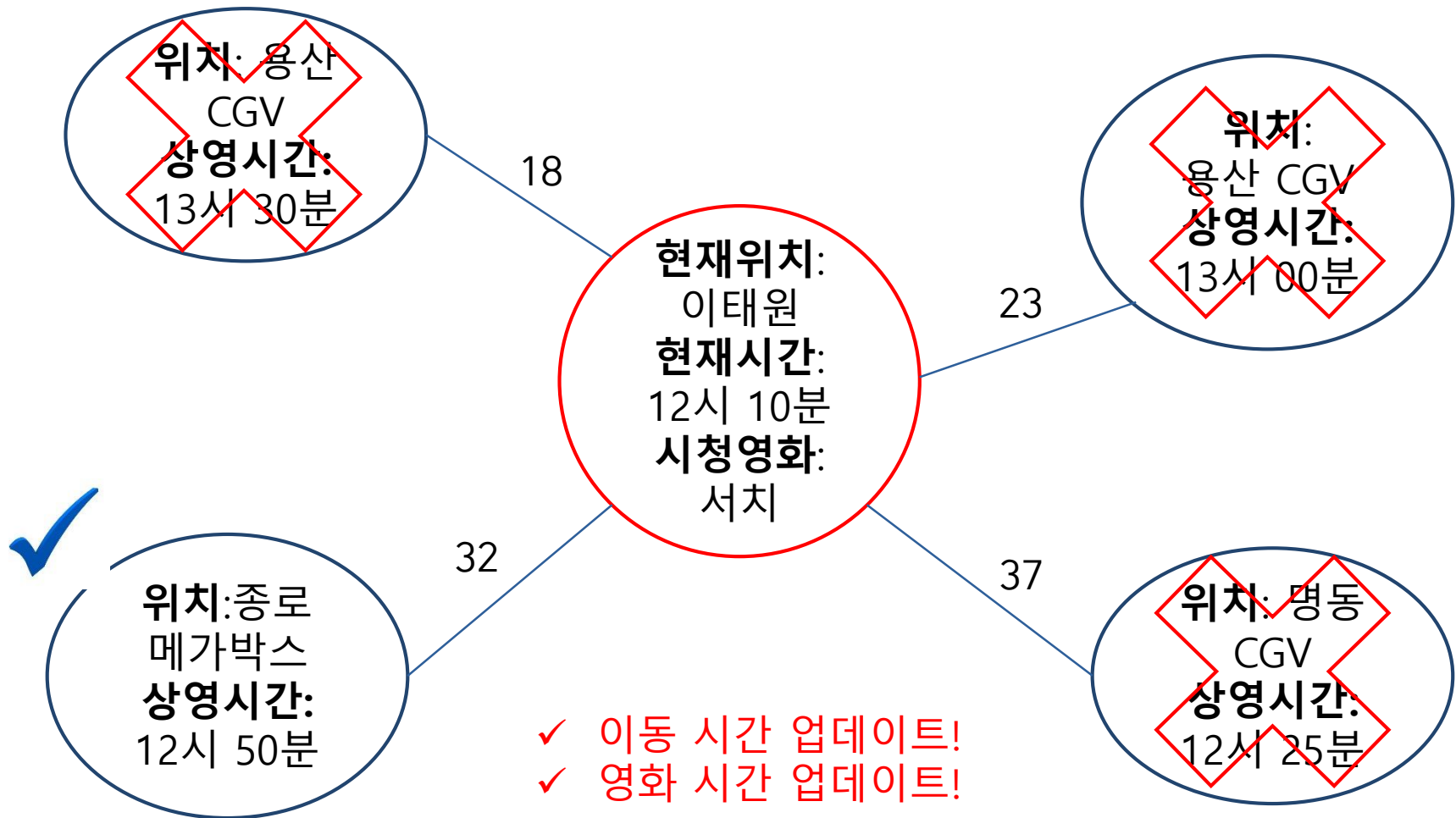
- ✓ 인접 리스트를 사용하여 영화관의 상영영화와 시간을 저장합니다.



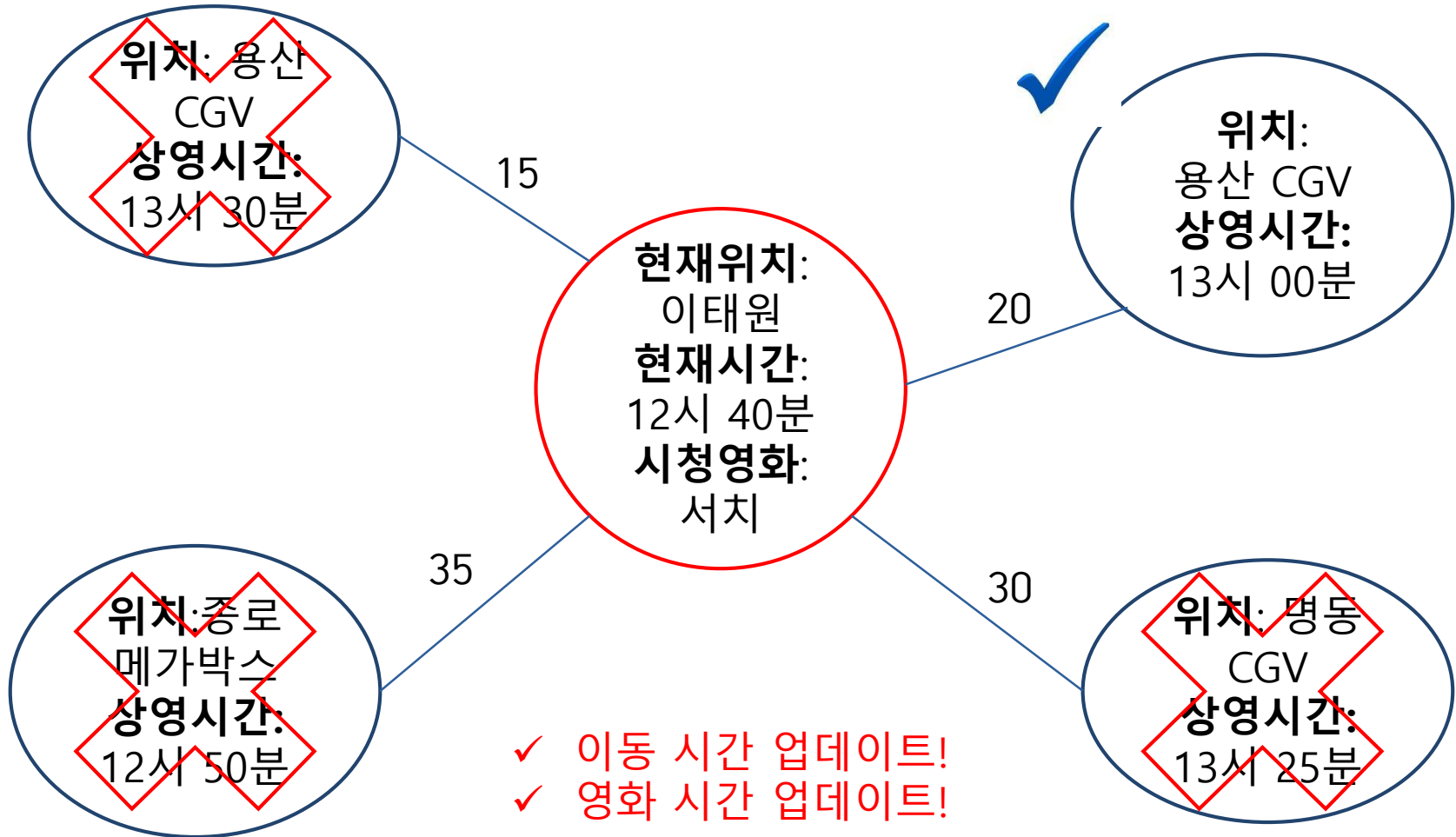
# 그래프 결과

	소요산	동두천	보산	동두천 중앙	...	...	...
소요산	0	2	4	6			
동두천	2	0	2	4			
보산	4	2	0	2			
동두천 중앙	6	4	2	0			
...					0		
...							
...							

# 그래프 도식화 1



## 그래프 도식화 2



# Implementation

## <솔루션 적용>

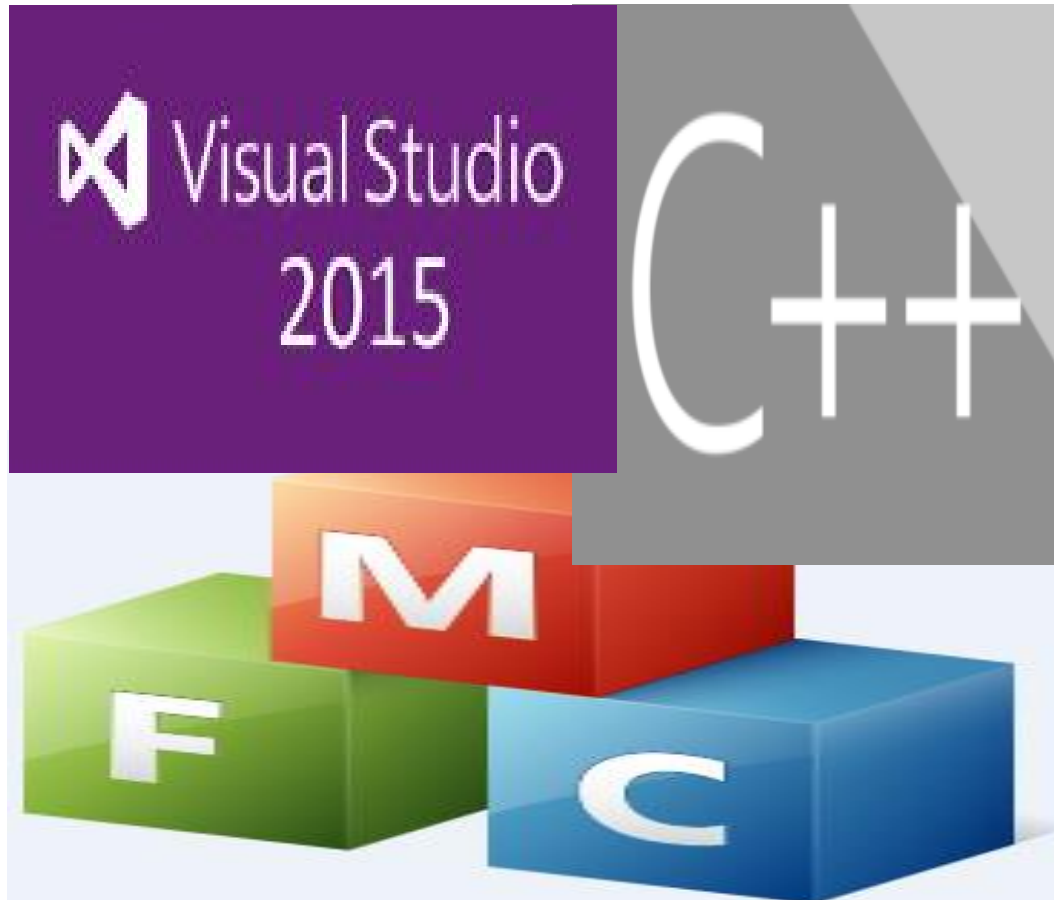
- 개발 환경
- 개발 과정

## <데이터 수집>

- 데이터 수집 환경
- 데이터 수집 과정
- 데이터 수집 출처



# 개발 환경



## 개발 과정

```
void Graph::DijkstraAlgorithm(int start){
    // start : 시작점 -> 역의id
    int pos;
    int min;

    for (int i = 0; i < MAX_STATION; i++)distance[i] = M;

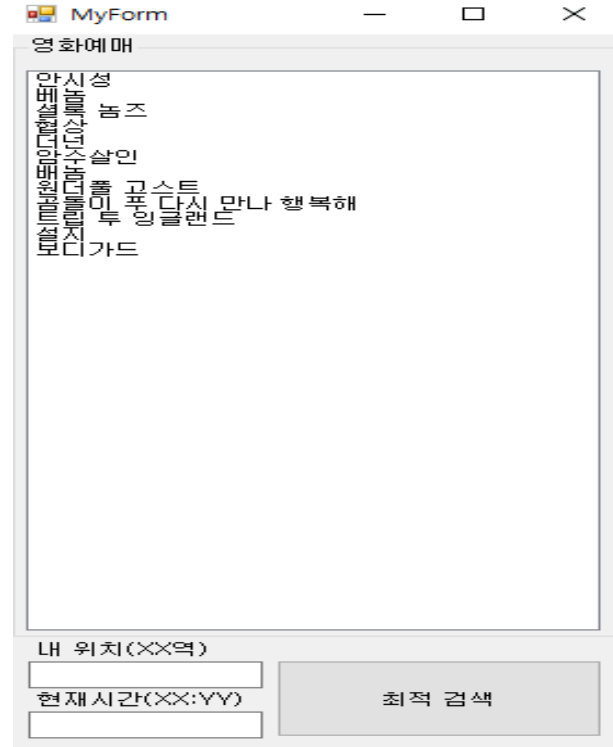
    for (int i = 0; i < MAX_STATION; i++){
        distance[i] = map[start][i];
    }
    bool check[MAX_STATION] = { false, };

    check[start] = true;

    for (int i = 1; i < MAX_STATION; i++){
        min = M;
        for (int j = 1; j < MAX_STATION; j++){
            if (!check[j] && min > distance[j]){
                min = distance[j];
                pos = j;
            }
        }
        check[pos] = true;

        for (int j = 1; j < MAX_STATION; j++){
            if (distance[j] > distance[pos] + map[pos][j]){
                distance[j] = distance[pos] + map[pos][j];
            }
        }
    }
}
```

- 출발역으로 설정한 역부터 모든 역까지의 최단거리를 다익스트라 알고리즘으로 배열 형태로 구현  $O(n^2)$
- Floyd algorithm :  $O(n^3)$
- Bellman-Ford algorithm : 음수치 필요x



- MFC를 활용하여 간단한 UI 구성

# 데이터 수집 환경



# 데이터 수집 과정

[illegible]

A	B	C
CGV 강남 안시성		6:30
CGV 강남 안시성		8:50
CGV 강남 안시성		11:25
CGV 강남 안시성		14:05
CGV 강남 안시성		16:30
CGV 강남 안시성		19:00
CGV 강남 안시성		21:30
CGV 강남 안시성		0:00
CGV 강남 안시성		2:30
CGV 강남 안시성		4:50
CGV 강남 배움		6:40
CGV 강남 배움		8:20
CGV 강남 배움		9:25
CGV 강남 배움		11:05
CGV 강남 배움		12:15
CGV 강남 배움		13:55
CGV 강남 배움		15:15
CGV 강남 배움		16:45
CGV 강남 배움		18:15
CGV 강남 배움		19:50
CGV 강남 배움		21:15
CGV 강남 배움		22:35
CGV 강남 배움		0:25
CGV 강남 배움		1:35
CGV 강남 배움		3:25
CGV 강남 배움		4:40
CGV 강남 설록 놀즈		11:10
CGV 강남 설록 놀즈		1:55
CGV 강남 협상		7:30
CGV 강남 협상		10:10



subway - 메모장			
파일(F)	편집(E)	서식(O)	보기
19	20	2	월
20	21	2	계
21	22	2	성
22	23	2	석
23	24	3	식
24	25	2	대
25	26	2	회
26	27	1	기
27	28	1	창
28	29	1	리
29	30	1	동
30	31	2	동
31	32	2	동
32	33	3	동
33	34	2	동
34	35	3	동
35	36	2	동
36	37	3	동
37	38	3	동
38	39	2	동
39	40	2	동
40	41	2	동
41	42	2	동
42	43	3	동
43	44	2	동
44	45	2	동
45	46	3	동
46	47	1	동
47	48	2	동
48	49	2	동
49	50	2	동
50	51	2	동
51	52	1	동
52	53	2	동
53	54	2	동

# 데이터 수집 출처

- [www.cgiv.co.kr](http://www.cgiv.co.kr)
- [www.megabox.co.kr](http://www.megabox.co.kr)
- [www.lottecinema.co.kr](http://www.lottecinema.co.kr)
- <http://www.seoulmetro.co.kr/kr/cyberStation.do?menuIdx=538&action=info>

Q&A

감사합니다.

#이용

#01040871966

#kstuve12@naver.com