2016년도 **인문자연탐사 보고서**

-삶, 세종시를 탐구하다-

세종시의 인구 밀도와 지도 분석을 통한 최적의 소방서 위치 조건 탐구

2016. 10. 22

송민근, 양현규, 오영환, 윤종빈, 우지원

Sejong Academy of Science and Arts
세종과학예술영재학교

세종시의 인구 밀도와 지도 분석을 통한 최적의 소방서 위치 조건 탐구

송민근, 양현규, 오영환, 윤종빈, 우지원

1. 탐사의 필요성

가. 탐사 동기

경향신문에 '화재·교통사고 가장 불안 세종시, 소방인력 최저'라는 제목의 글이 올라왔다. 글의 내용은 세종시의 교통사고, 화재 안전지수에서 최하위 등급을 받은 데 이어 소방 대비 인력이 평균보다 56% 부족하다는 기사였다. 따라서 이러한 문제를 해결하기 위해서는 근본적으로 소방인력 확충이 필요하나, 소방서의 적절한 배치로 적은 소방력임에도 불구하고 최대한의 효율을 낼 수 있는 방법 또한 필요하다고 판단했고, 소방서가 최대한의 효율을 낼 수 있는 하나의 방법으로 효율적인 소방서의 위치가 될 수 있는 장소를 모색하기로 했다.

나. 탐구 목적

인구 밀도, 사건 사고 발생 비율 등을 고려하여 가장 효율적으로 화재 진압 및 구급 구조 활동에 임할 수 있는 '적절한 소방서의 위치'가 될 수 있는 곳을 파악한다. 이후 실제 소방서의 위치와 우리가 산정한 소방서의 적정 위치를 비교한다.

다. 접근 과정

소방서의 효율(능률)이란, 현장 도착 시간을 기준으로, 얼마나 빠르게 사태를 초기진압 하느냐에 대한 것을 말하는 것이다. 교통 상황(접근성)과 위치적 조건 등이 가장 큰 요인으로 고려된다. 예를 들어, 사건 발생 후 현장 도착 시간이 대부분의 경우에서 적게 걸리는 위치가 효율적인 위치가 되는 셈이다.

위치 조건의 기준은 '인구 밀도'와 '사고 발생률'을 고려한 '현장 도착 시간'을 기준으로 한다. 큰 인명피해 혹은 잦은 사고발생 지역에 대한 빠른 대책이 필요하기 때문이다.

인구 밀도와 사고 발생률이라는 두 요인에 대해서, 사람이 많으면 그에 따른 사고 발생률이 높을 것으로 예측한다. 즉, 두 요인은 서로 비례관계에 있을 것으로 가정한다. 이 때 유동인구가 높은 곳(상가 등), 혹은 산업 단지 등의 경우 인구 밀도와 큰 상관 없이 사고 발생률과 사고 발생 시 피해 규모가 클 것으로 생각된다. 이 경우에 대해서는 추가적인 사고 발생률(혹은 위험도)을 고려한다.

2. 탐사 과정

가. 진행한 탐사 경로(일정)

1) 탐사 목적 설계 및 자료 조사
2) 각 생활권 별 인구수 및 면적, 인구 밀도 등 소방서 위치를 결정하는데 필요한 각종 수치를 정부 보고서를 통하여 구함.
3) 현재 행정중심복합도시의 안전센터 세 곳에 가서 주변 환경을 파악한다.
4) 한국토지주택공사에서 탐구 방법과 자료 조사에 대한 자문을 구한다.
5) 세종시의 소방서 배치 방법을 어떠한 기준을 통해서 결정할지 논의 - 시설에 따른 중요도, 현장 도착 시간을 고려해서 설정하기로 함.
6) 세종시(행정중심복합도시) 내의 소방서 수가 얼마나 필요할지 계산.
7) 앞서 계산한 내용을 바탕으로 가장 적당한 소방서의 위치를 결정한다.

나. 탐구 내용

1) 소방서에 대한 분류

대한민국의 소방서는 크게 총 세 가지 형태로 구성되어있다. 이는 소방관서, 119 안전센터, 119 지역대로 나뉜다. 소방관서는 행정업무(소방행정, 예방검사교육, 구조구급통계, 신고접수 등의 업무를 수행한다. 119 안전센터는 화재·구급업무에 대응하는 역할을 한다. 119 지역대는 소방수요가 있으나 119 안전센터를 설치할만한 수요가 아닌 경우에, 119 안전 센터 산하에 지역대가 추가 배치된다.

다음 보고서에서는 특수한 경우를 제외하고, 119 안전센터와 119 지역대를 구분하지 않고 모두 '소방서'로 통칭하기로 한다.

2) 필요 소방서의 수 계산

정부 보고서인『소방력 배치기준에 관한 연구』의 내용에 의거하여, 현재 세종시 내에 필요한 소방서의 수를 계산한다. 다음 보고서에 의하면, 일본의 1개 소방서 관할 담당 면적에 대해서 [그림 1]과 같은 식을 유도할 수 있다.

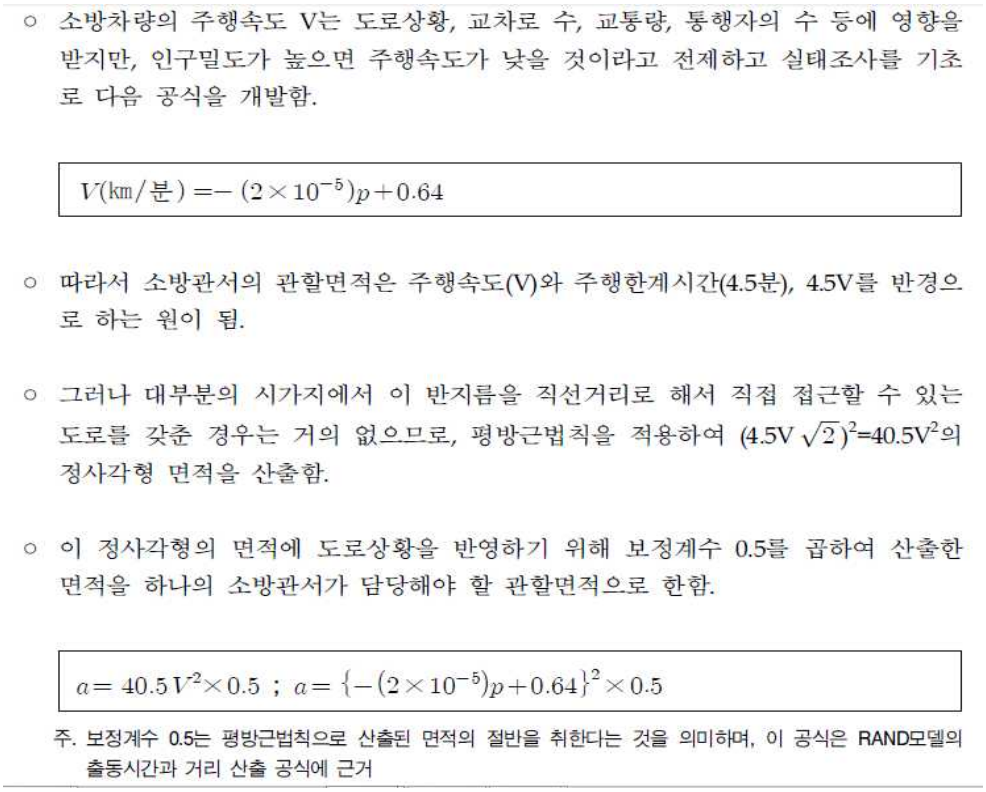
세종시 중 행정중심복합도시의 넓이=$72.91km^2$, 인구수= 500,000명.

$$\therefore p(\text{인구밀도}) = 6857.8(\text{명}/km^2)$$

$$a = 40.5(-2\times10^{-5}\times p + 0.64)^2 \times 0.5 = 10.2$$

따라서, 1 개의 세종시 소방서가 관할하는 면적은 $10.2km^2$ 이다.

$$\text{소방서 수} = \frac{\text{도시 계획 면적}}{\text{소방서 당 담당면적}} = \frac{72.91}{10.2} \simeq 7$$



[그림 1. 소방관서 당 관할 면적]

결론적으로, 행복중심복합도시를 기준으로 할 때, 필요한 소방서의 수는 7개가 필요하다는 계산이 나온다. 이 때 한 소방서가 부담할 수 있는 면적은 $10.2km^2$, 거리는, 현장 도착 시간 4.5분을 고려하여서 직선거리로 약 $2.5km$이다.

3) 소방서 적절한 위치를 판단하는 요인

일단, 우리가 가장 중점을 두고자하는 교통 즉, 접근성의 측면에서 바라보자. 우리는 행정중심 복합 도시 내부에서의 접근성을 길의 면적으로 접근해보기로 하였다. 도시의 전체 지도를 크기가 같은 격자 구간으로 나누어 그 안에 속하는 길의 면적을 측정했을 때, 길의 면적이 넓다는 것은 아마도 그 곳의 교통이 편리하고 접근성이 좋다는 것을 뜻한다고 예상했다. 이러한 이유로, 구간 중 길의 면적이 넓은 구간에 대해 가중치를 부여했다.

둘째로, 우리는 인구밀도에 대해서 고려했다. 인구밀도는 일반적으로 화재 발생 빈도와 직접적 관계를 가질 것이다. 인구밀도가 높은 구간에 대해서는 가중치를 부여했다.

셋째로, 화재 발생 빈도(위험성)에 대해 고려했다. 이는 위에 말했듯, 인구 밀도와는 관계없이 상가처럼 거주 인구는 적지만, 유동 인구가 많다거나 산업 단지처럼 건물 자체의 화재 위험성이 높은 경우에 대해 고려하는 요소이다. 우리는 화재 발생 빈도와 인구 분포를 바탕으로 어떠한 지역의 어떠한 건물이 이러한 위험성을 갖고 있는지 판단해 위험 건물이 위치하는 구간에는 가중치를 주었다.

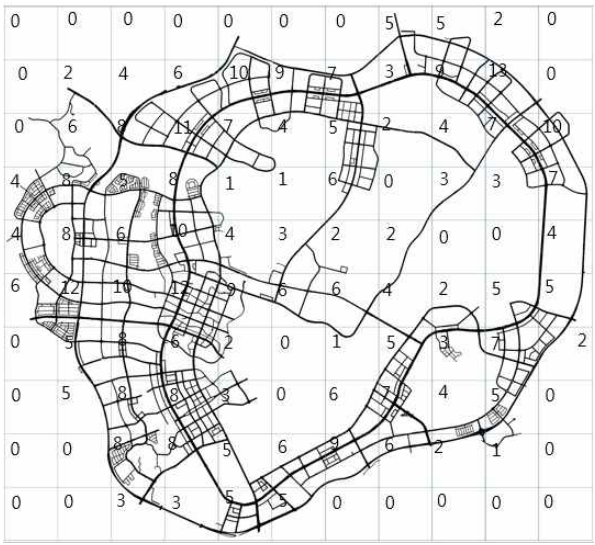
넷째로, 현장 답사를 통해 이 모든 가중치들을 계산해 도출되는 행정 중심 복합 도시의 최적

소방서 위치에 과연 실제로 소방서가 위치해서 좋은 역할을 다할 수 있는지 알아보고자 한다.

3. 탐구 결과

가. 소방서 위치를 결정하는 요인에 대한 고려

우리는 다음 그림과 같이 세종시를 11*10 격자로 나누어 각 가중치 값을 입력하고 그에 대한 총 합을 통하여 최종 소방서 위치로 적합한 장소를 판단할 것이다.



[그림 2. 세종시 분할(예시)]

이 때, 인구의 밀도는 각 생활권 별 인구밀도자료(출처: 행정중심복합도시건설 개발계획변경(2010.7))를 통해서 구하였으며, 각 생활권에서 인구는 모두 고르게 분포한다는 가정 하에서 인구 밀도 값을 격자에 대입하였다. 만약 각 격자에서 생활권이 겹치는 부분이 존재한다면, 그 격자 안에 겹친 지역의 비율만큼 인구 밀도 값을 대입하였다. 예를 들어, 인구밀도가 100인 지역과 40인 지역이 6:4 비율로 들어간 격자가 있다면, 그 격자에는 $100\times0.6+40\times0.4=76$ 76의 값이 들어가는 식이다.

연기면	14	13	0	0	1	92.86	0	0	7.14	0	0	0	22,686	11,896	10,790	5	80
한솔동	2	2	0	0	0	100	0	0	0	0	0	0	1,125	982	143	1	5

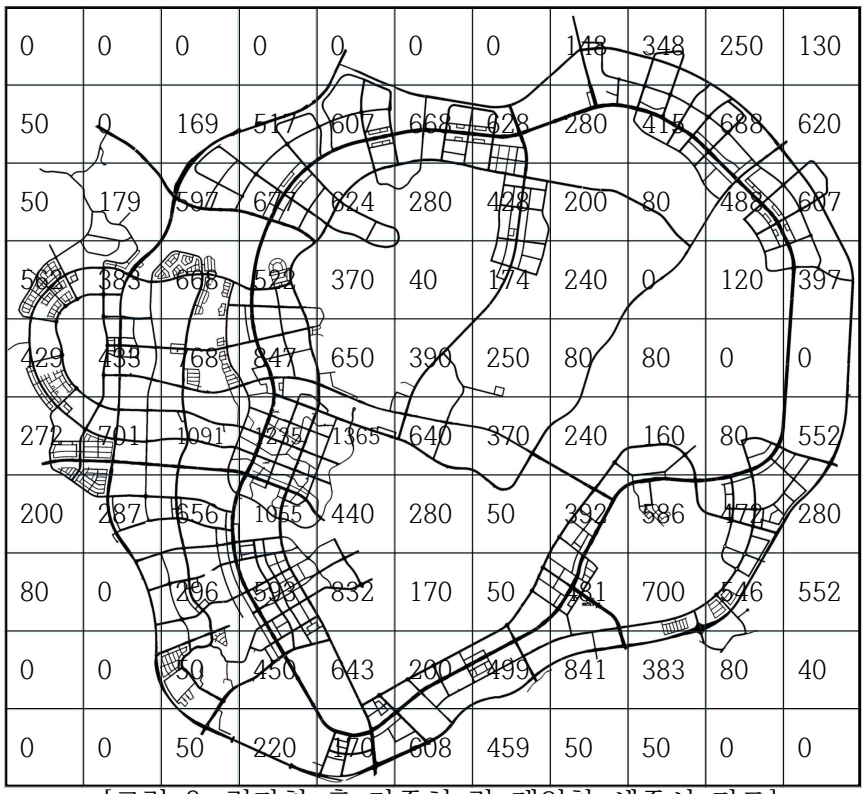
인구수가 훨씬 많은 한솔동에 비해 인구수가 적은 연기면의 2015년 10월 1일부터 2016년 9월 30일 까지 1년간 화재 발생 횟수는 압도적임을 볼 수 있다. 이는 연기면에 거주인구는 적지만 유동인구가 많은 시청, 호수공원, 은하수 공원 등이 있기 때문이라고 보여진다. 즉, 우리는 유동인구가 많은 공원, 영화관, 대형마트, 공장, 상가단지에 대해 가중치를 부여할 것이다.

교통에 대한 가중치는 각 구간에서 다른 구간으로 가는 길의 수를 세는 방식으로 산정되었

다. 차선의 수가 많은 대로의 경우, 다른 좁은 도로와는 다르게 교통이 상당히 용이하기 때문에 두 도로에 대해 차이를 두었다. 6차선 이상의 대로에는 2의 점수를, 4차선 이하의 도로에는 1의 점수를 부여하였다. 또한, 유동인구(사고 다발 위험지역)에 대한 가중치는 유동인구가 많이 몰릴 대형 상가 혹은 위락시설이나 정부 기관 시설 중심으로 하여, 해당 시설이 존재하는 격자에는 가중치를 2, 그 주변 격자에는 가중치를 1 부여하는 방식으로 진행되었다.

다음 세 가지 방식으로 진행된 각 격자의 가중치 값에서, 우리는 세 요인에 대한 변수 값을 비슷한 비율로 산정하기로 하였다. 그 이유는, 소방서의 위치를 판단하는 세 가지 요인, 인구 밀도와 교통의 용이성, 위험도 가중치 중 어떤 요인이 더 우선되어야 하는지, 얼마나 더 우선되어야 하는지 정확한 사전 연구 결과나 시뮬레이션 결과를 얻지 못하였기 때문이다. 그래서, 각기 다른 방식으로 산정한 이 세 요인들이 최종 가중치 합산 과정에서는 결국 모두가 비슷한 비율로 산정되도록 결정하였다. 이 세 요인의 가중치 값을 합쳐서 산출한 최종 가중치 값은 다음 식을 통해 산정되었다.

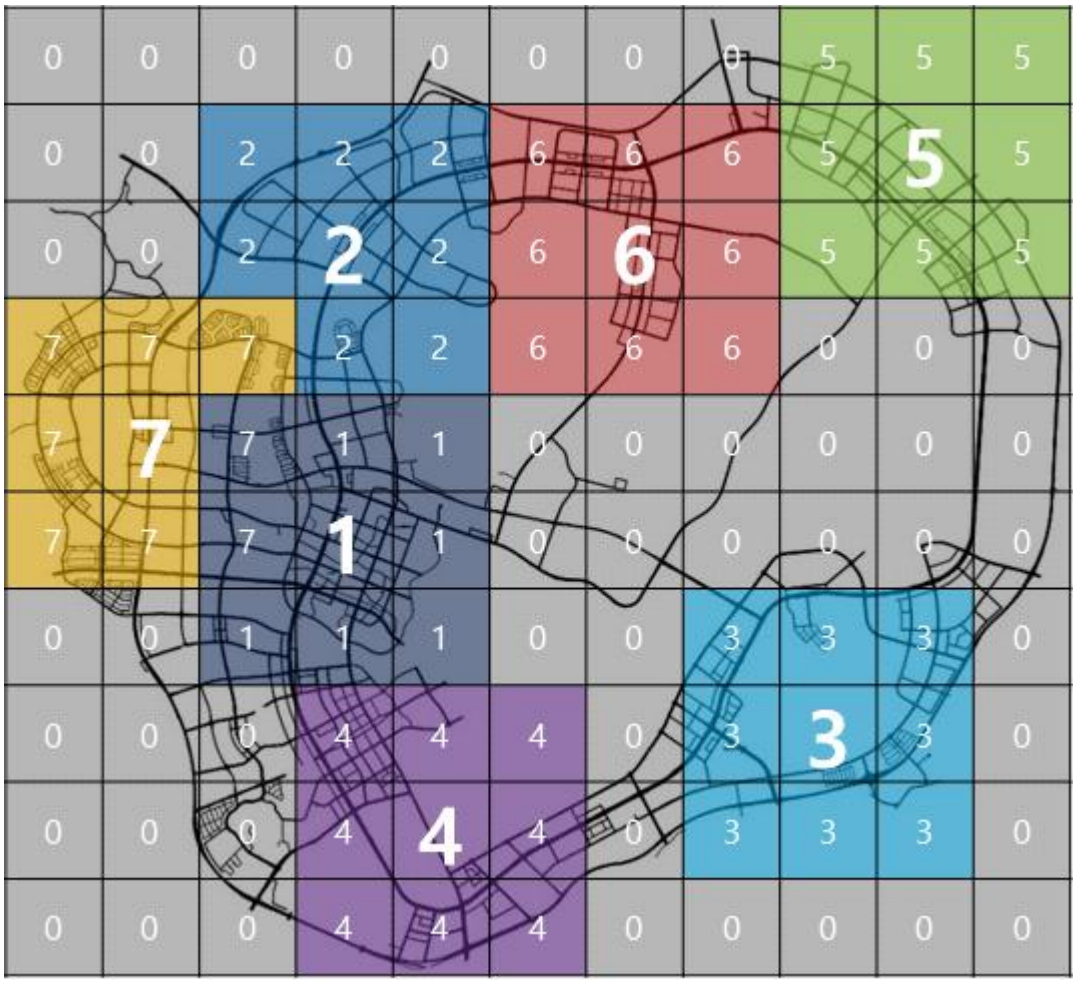
최종 가중치 = 인구 밀도 + 교통*40 + 위험도 가중치*50



[그림 3. 격자화 후 가중치 값 대입한 세종시 지도]

위 식을 통해 각 격자의 값을 구한 다음 소방서의 위치를 구하기 위해 코딩을 이용했다. 사용한 코드는 보고서 맨 끝에 별첨을 통해 첨부하도록 한다. 이를 통해 소방서의 위치가 될 구역을 구했다. 아래 그림에서, 각각의 사각형은 하나의 소방서가 담당하는 담당 구역을

가리킨다. 이 지도에서 소방서의 위치는, 각 사각형의 중심 혹은, 그와 근접한 교통이 용이한 곳. 즉, 굵은 글씨가 씌여진 사각형의 범위 내에 위치하게 된다. 예를 들어, 1과 같은 경우, 다음의 위치(정부 청사 부근)에 소방서가 위치하는데, 4와 같은 경우 가장 중심에 있는 사각형의 범위가 강에 위치하고 있기 때문에, 강이 아닌 한솔동의 다리 부근 강가에 소방서가 위치해야 될 것이다.



[그림 4. 소방서의 담당 구역을 격자화하여 나타낸 세종시 지도]

4. 결론 및 제언

가) 결론

따라서 이 탐구의 결과가 될 세종시 내 적정 소방서의 수는 7개, 그 소방서 각각의 위치는 [그림 4]에 나와있는 대로 특정 범위를 이야기 할 수 있다.

결론 도출 이후, 우리는 탐구의 결과와 실제 소방서의 위치를 비교해 보았다. 비교 결과, 7의 위치에는 아름119 소방센터와 세종소방서가, 1의 위치에는 어진119소방센터가, 4의 위치에는 한솔119소방센터가 위치하는 것을 확인함으로서, 실제 도시 계획 상의 소방서 배치와, 우리 연구 상의 소방서 배치에 큰 차이가 없음을 확인할 수 있었다. 나머지 지역(4,5,6 생활권)의 경우는 현재 개발중인 구역으로서, 아직 소방서가 지어지지 못하였다는 점을 고려하면, 우리의

연구 결과는 나름 성공적이었다고 평가할 수 있다.

나) 제언

우리 연구에서 사용되었던 자료들은 모두 '국가 계획 상, 2030년까지 목표하는 수치'를 기준으로 작성된 보고서를 기준으로 했기에, 실제 값을 정확히 반영하지 못했다. 또한 세밀한 수치적 자료의 부족으로 격자의 크기를 상당히 크게 잡았기에, 정밀한 자료가 존재하여 좀 더 세밀한 격자를 구성하였다면, 결과의 정확성을 조금 더 높게 잡을 수 있었을 것이다.

이외에도 지형, 풍속, 풍향, 교통체증 심화구간과 같은 다양한 변수를 자료 부족 및 시간 부족으로 고려하지 못하였으며, 최종 가중치 산술식 또한 시뮬레이션 구상 시간이 부족하여 각 요인 간의 상관관계 혹은 소방서에 영향을 주는 정도에 대한 정밀한 파악이 이루어지지 않은 점이 아쉬운 부분이다.

5. 활동 후기

1410 양현규. 정밀한 연구를 하기에, 3일이라는 시간은 너무나도 부족한 시간이었다. 사실 세종시 내의 도로 길이를 전부 다 하나하나씩 재어가고, 각 블록당 가구 수, 유동인구 수, 교통량등을 전부 다 고려하여 정말 정밀한 값을 도출하고자 하였으나, 현실적인 문제와 자료 부족으로 인해 문제를 단순화시켜서 해결할 수 밖에 없었다. 그럼에도 불구하고, 인문자연탐사가 아니었다면 평생 알아보려고 하지 않았을 소방인력과 소방 기준에 관한 많은 논문, 보고서를 읽어보며 시야를 넓힐 수 있었다. 힘든 시간이었지만, 좋은 경험이 되었기에 만족스럽다고 생각한다.

1409 송민근. 평소 세종시에 그렇게 큰 관심을 가지고 있지 않았을 뿐더러 직접 발로 뛰어다니는 것을 별로 좋아하지 않는 편이었던 나였다. 하지만 하면 할수록 열심히 해야겠다는 마음이 생겼다. 그 발단이 된 첫 번째 원인은 주제 선정이었다. 전혀 예상치 못했지만 선배들이 내 관심 분야인 정보쪽을 지향했기 때문이었다. 물론 그것은 발단일 뿐이었다. 실제로 탐사를 뛰면서, 평소 신도시니 마냥 좋을 것이라고 생각했던 세종시의 아직 부족한 점들이 속속히 드러났기 때문이었다. 그러니 끝나고 느낀 점이라고 한다면, 세종시에 대해 관심이 높아졌기 보다는 역시 관심이 없더라도 여러 가지 우물을 파보는 것도 나쁘지 않다는 생각이 들었다.

2408 오영환. 이번 탐사는 저번보다 어려운 편이었다. 1학년 때에는 생태 조사 이상의 무언가를 하지 않았으나, 이번에는 지리와 관련된 것을 해 보니 훨씬 고려할 변수가 많고, 논리적인 해법을 알아낼 때까지의 시간이 몇 배는 더 걸린다는 것을 알게 되었다. 하지만 이런 활동이 끝나고 나니 저번보다 더 보람찬 기분이 들었고, 나중에도 한번 더 하고 싶다는 생각이 들었다.

2409 우지원. 나는 이 연구를 시작하기 전, 과연 우리가 소방서의 최적 위치를 찾는다는 복잡한 연구를 완성할 수 있을지 의문을 가졌다. 하지만, 연구를 시작하고 난 뒤 우리는 우리들 나름대로의 기준을 정했고, 그 기준에 맞추어 자료를 조사하고 정리했다. 그리고 우리는 그들 사이의 규칙을 찾았고, 드디어 우리가 원하던 결론에 도달할 수 있게 되었다. 높고 어려운 벽으로만 보였던 연구가 쉽게 진행되는 것을 보며 나는 나의 수준이 한층 더 향상됨을 느꼈고, 즐거움을 느꼈다. 나에게 이러한 기회를 준 인문 자연 탐사에 대해 감사를 표하는 바이다.

2410 윤종빈. 작년과는 다르게 금강 주변 자연환경을 그 근처에서 숙박하면서 탐사한 것이 아니라서 자연을 통해 느낀 감정이나 생동감은 떨어졌지만 인터넷,
전문가의 자문, 토의, 현장조사를 통해 연구를 진행하여 훨씬 더 전문적
이고 생산적인 연구를 진행했다고 생각이 된다. 앞으로도 이러한 연구를
꾸준히 진행하였으면 좋겠다.

6. 참고 문헌

- 『소방력 배치기준에 관한 연구』 - 정책연구관리시스템
- 『행정중심복합도시건설 개발계획변경(2010.7)』 - 행정중심복합도시건설청

7. 부록 - 결과를 구하기 위해 사용한 코드

```
// include~
#include<stdio.h>

// visual studio scanf error 무시 명령어
#pragma warning(disable:4996)

// 매크로
#define FOR(i,n) for(int i=1;i<=(n);i++)      // 초기값이 1인 for문
#define FOR2(i,k,n) for(int i=(k);i<=(n);i++) // 초기값이 k인 for문
#define MAX 1001

// 구조체
typedef struct coordinate_system {
	int x, y;
```

```
	int cost;
}Csystem;

// 전역 변수
int n, m; // 세로 격자 개수, 가로 격자 개수
int s[MAX+100][MAX+100]; // 가중치(중요도) 배열
int n_station; // 소방서 개수
int station_size; // 관할 영역의 변 길이

void input() // 입력 함수
{
	int i,j;

	scanf("%d %d", &n, &m);
	FOR(i, n)
	{
		FOR(j, m)
		{
			scanf("%d", &s[i][j]);
		}
	}
	scanf("%d", &n_station);
	scanf("%d", &station_size);
}

Csystem ans[101];		// 답의 각 소방서 위치가 적힐 배열
int cnt = 0;
void make_zero(int x, int y);
int cal_effi_in_size(int x, int y);

void update()
{
	int i,j;
	int t;

	FOR(t, n_station)
	{
```

```
            Csystem max_ans;
            max_ans.cost = -1;

            FOR(i, n)
            {
                    FOR(j, m)
                    {
                            int effi_cost = cal_effi_in_size(i, j);
                            if (effi_cost > max_ans.cost)
                            {
                                    max_ans.x = i;
                                    max_ans.y = j;
                                    max_ans.cost = effi_cost;
                            }
                    }
            }

            ans[t] = max_ans;
            make_zero(max_ans.x, max_ans.y);
      }
}           // 프로그램 해결 함수

int print_check[MAX][MAX];      // 출력 전용 배열
void output() // 출력 함수
{
      int i,result=0;
      int j, k;
      printf("줄/칸/가중치\n");
      FOR(i, n_station)
      {
            printf("%d      %d      %d\n",      ans[i].x+(station_size-1)/2,
ans[i].y+(station_size-1)/2, ans[i].cost);
            result += ans[i].cost;
            print_check[ans[i].x][ans[i].y] = 1;
            FOR2(j, ans[i].x, ans[i].x + station_size - 1)
            {
                    FOR2(k, ans[i].y, ans[i].y + station_size - 1)
```

```
                {
                        print_check[j][k] = i;
                }
        }
    }
    printf("\n총 가중치 합 = %d\n\n", result);
    printf("관할 총 영역\n");
    FOR(i, n)
    {
        FOR(j, m)
        {
            printf("%d ", print_check[i][j]);
        }
        printf("\n");
    }
}

int main()
{
    freopen("input.txt","r",stdin);
    freopen("output.txt", "w", stdout);

    input();
    update();
    output();

    return 0;
}     // main 함수

/*------------------ update 전용 함수 --------------------*/

int cal_effi_in_size(int x, int y)
{
    int i, j;
    int sum = 0;
    FOR2(i, x, x + station_size - 1)
    {
```

```
                FOR2(j, y, y + station_size - 1)
                {
                        sum += s[i][j];
                }
        }
        return sum;
}
void make_zero(int x, int y)
{
        int i, j;
        FOR2(i, x, x + station_size - 1)
        {
                FOR2(j, y, y + station_size - 1)
                {
                        s[i][j] = 0;
                }
        }
}
```