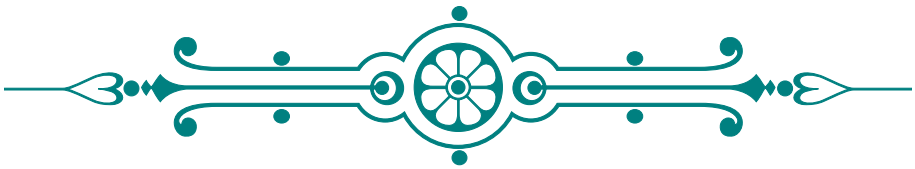


컴퓨터 알고리즘과 실습

학번 : 2016110056

학과 : 불교학부

이름 : 박승원

날짜 : 2017년 3월 29일

문제 1 (25점)

- 최소 최대값 동시 찾기 문제
- 1~100000범위내에 있는1000개의 랜덤한 양의 정수를 생성한 후
  - 수업시간에 배운 최소값 찾기 Minimum() 함수와 이를 응용한 Maximum()함수를 이용하여 최소값과 최대값을 찾고 이를 출력하여라.
  - 수업시간에 배운 FindMinMax() 함수를 이용하여 동시에 최소값과 최대값을 찾고 이를 출력하여라.

  주의) 각 언어에 있는 패키지나 기본 함수를 사용하면 안되고 직접 함수를 구현해서 사용해야 함.

```
// 2016110056   박승원
#include<iostream>
#include<random>
using namespace std;

int Minimum(int* p) {
    int min = 10000000;
    for(int i=0; i<1000; i++) if(p[i] < min) min = p[i];
    return min;
}

int Maximun(int* p) {
    int max = 0;
    for(int i=0; i<1000; i++) if(p[i] > max) max = p[i];
    return max;
}

pair<int, int> FindMinMax(int* p) {
    int min = 10000000, max = 0, small, large;
    for(int i=0; i<1000; i+=2) {
```

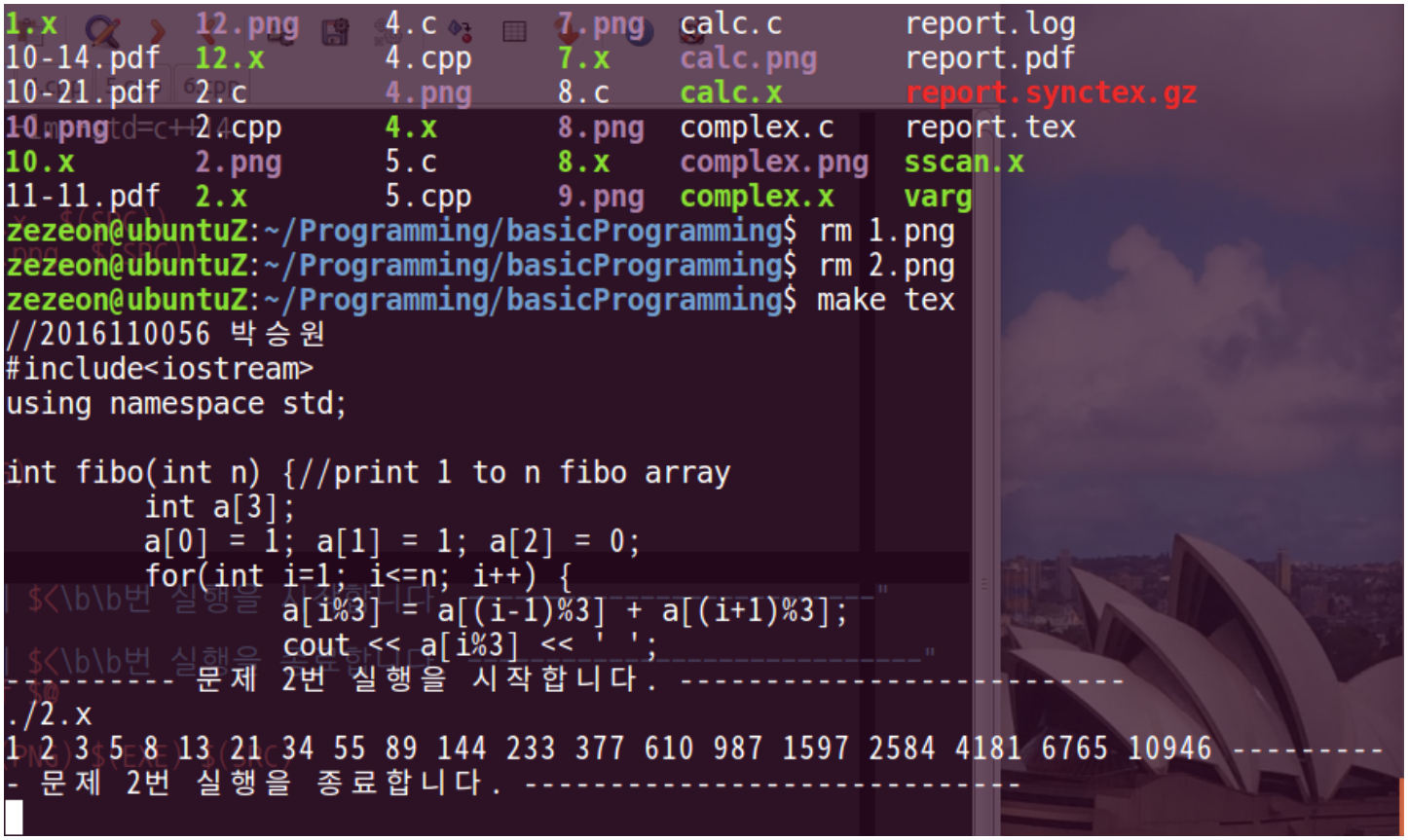
```
        if(p[i] < p[i+1]) {
            small = p[i];
            large = p[i+1];
        } else {
            large = p[i];
            small = p[i+1];
        }
        if(small < min) min = small;
        if(large > max) max = large;
    }
    return {min, max};
}


int main(int ac, char** av) {
     uniform_int_distribution <> di(1, 100000);
    random_device rd;

    int ar[1000];
    for(int i=0; i<1000; i++) ar[i] = di(rd);
    cout << Minimum(ar) << ' ' << Maximun(ar) << ' ' << endl;
    auto a = FindMinMax(ar);
    cout << a.first << ' ' << a.second << endl;
}
```

```
d4.pfb></usr/share/texlive/texmf-dist/fonts/type1/public/nanumtype1/nanummjmd5.
pfb>
Output written on report.pdf (8 pages, 2585371 bytes).
SyncTeX written on report.synctex.gz.
Transcript written on report.log.
evince report.pdf
zezeon@ubuntuZ:~/Programming/basicProgramming$ make tex
g++ 1.cpp -o 1.x -g -fmax-errors=1 -lm -std=c++14
//2016110056 박승원
#include<iostream>
#include<random>
using namespace std;

int Minimum(int* p) {
        int min = 10000000;
        for(int i=0; i<1000; i++) if(p[i] < min) min = p[i];
        return min;
}
---------- 문제 1번 실행을 시작합니다. ----------
./1.x
2 99957
2 99957
---------- 문제 1번 실행을 종료합니다. ----------
```

문제 2

- 피보나치 수열 함수를 구현해 보고 F_1부터 F_{20}까지를 출력해 보아라.

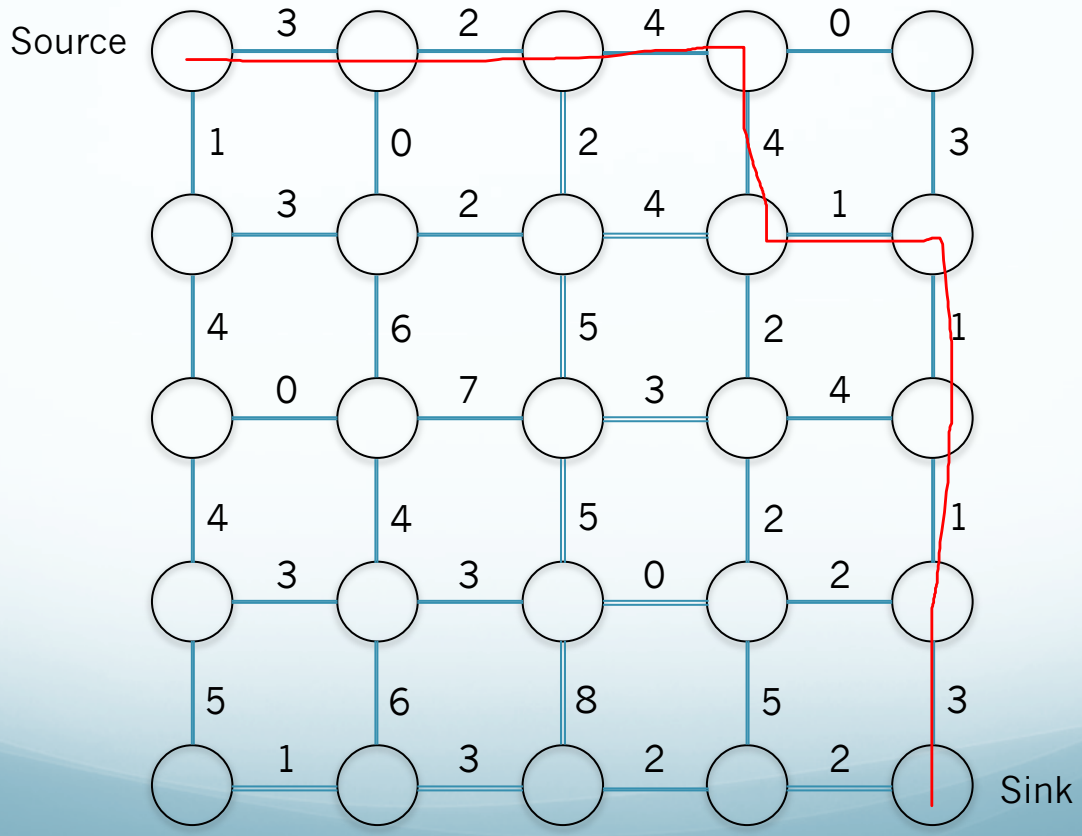
```
// 2016110056   박승원
#include<iostream>
using namespace std;

int fibo(int n) {// print 1 to n fibo array
    int a[3];
    a[0] = 1; a[1] = 1; a[2] = 0;
    for(int i=1; i<=n; i++) {
        a[i%3] = a[(i-1)%3] + a[(i+1)%3];
        cout << a[i%3] << ' ';
    }
}

int main()
{
    fibo (20) ;
}
```



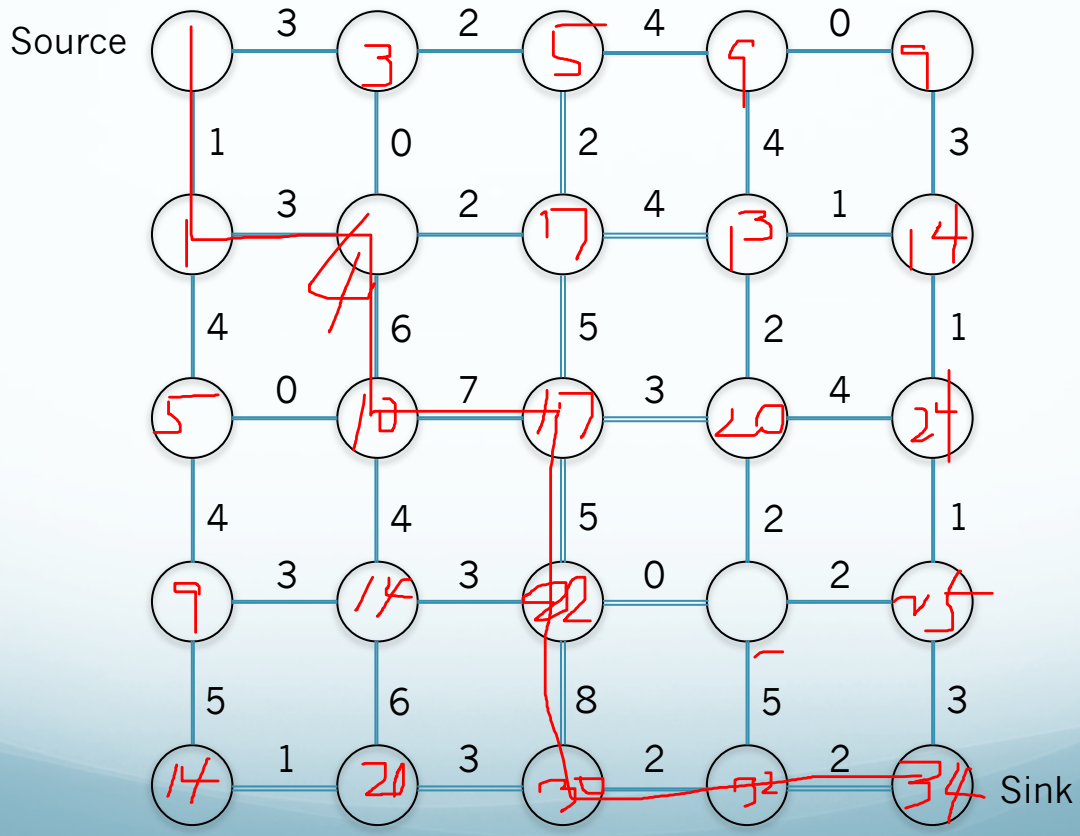
문제 3 (50점)

- Manhattan Tourist Problem
  - Manhattan Tourist Problem의 greedy 경우 경로를 구하시오. (10점)
  - Manhattan Tourist Problem의 Dynamic Programming 경우 경로를 구하시오. (10점)
  - Manhattan Tourist Problem의 Dynamic Programming 유사코드를 작성하시오. (30점)

Greedy Algorithm



Dynamic Programming



대각선으로 각 지점에 도달하는데에 걸리는 에지의 웨이트를 합하여 최고값을 쓴다. 대각선을 도착지점을 향하여 전진시키며 위를 반복한다. 다익스트라 알고리즘의 경우 음수로 웨이트를 변형하여 계산을 해보았으나, 다익스트라 알고리즘은 음수가 아닌 경우에만 최소 경로를 찾을 수가 있었다. 롱기스트 패스를 찾는 일반적인 문제는 NP완전문제로 아직까지 그 해법이 알려지지 않았다.

for i 1 to 5 : for j 1 to 5 :

$S_{i,j} \leftarrow max(\overrightarrow{W}_i + S_{i-1,j}, S_{i,j-1} + W_j\downarrow)$