

## CSE3081/AIE3051 (1반): 알고리즘 설계와 분석

### [HW 1] Solving Maximum Sum Subarray Problem

담당 교수: 임 인 성

2025년 9월 22일

마감: 10월 1일 수요일 오후 8시 정각

제출물, 제출 방법, LATE 처리 방법 등: 조교가 과목 게시판에 공지함.

**목표:** 이번 숙제는 Maximum Sum Subarray 문제에 대한 알고리즘을 구현한 후, 서로 다른 크기의 입력 데이터에 대한 수행 시간을 측정하여, 이론적인 시간 복잡도와 실제 수행 시간 간의 연관 관계를 분석해 봄을 목표로 한다.

#### 문제

- 다음과 같은 Maximum Sum Subarray 문제 (2D)를 고려하자.

Given a 2D integer array of size  $m \times n$ , find the maximum sum subarray **with at least one element**.

이 문제를 해결해주는  $O(mn^2)$ 의 시간 복잡도를 가지는 알고리즘을 구현하라. 이 구현은 자신이 찾은 subarray 영역의 원소들의 총합뿐만 아니라 그에 해당하는 subarray의 위-왼쪽 모서리와 아래-오른쪽 모서리 원소들의 인덱스를 찾아주어야 한다.

- 이번 숙제의 목적은

- (a) 해당 문제에 대하여 정확한 답을 구하는 효율적인 구현물을 산출할 뿐만 아니라,
- (b) 서로 다른 크기 ( $m, n$ )을 가지는 입력 영상에 대하여 수행 시간을 측정한 후,
- (c) 과연 그러한 수행시간이 이론적인 시간 복잡도와 일치하는지를 확인하는 것이다.

#### 입출력 방식

- 이번 숙제에서 구현하는 프로그램은 Visual Studio 솔루션의 프로젝트 디렉토리의 Data 디렉토리에 존재하는 **이름이 HW1\_config.txt인 텍스트 파일**에서 먼저 테스트 케이스의 개수를 읽어들이고, 한번에 한 줄씩 (i) bin 타입의 입력 데이터 파일 이름과 (ii) txt 타입의 결과 값 출력 파일 이름을 순서대로 읽어 들인 후, (iii) 그에 대한 계산을 수행하여 출력 파일에 계산 결과를 출력해주어야 한다. 이때 입출력 파일은 모두 위의 Data 디렉토리에 존재하여야 하며, 다음은 HW1\_config.txt 파일의 예이다.

```
2
dataset_1_1024x1024.bin dataset_1_1024x1024.txt
dataset_4_2048x2048.bin dataset_4_2048x2048.txt
```

- bin 타입의 입력 데이터에는 다음과 같은 정보가 binary format으로 저장되어 있다.

- (a) 첫 4 바이트에는 입력 array의 행의 갯수  $m$ 이 정수 타입으로 저장되어 있다.
- (b) 다음 4 바이트에는 입력 array의 열의 갯수  $n$ 이 정수 타입으로 저장되어 있다.
- (c) 다음  $4 \cdot m \cdot n$  바이트에는 입력 array의 원소가 row-major 순서로, 4 바이트에 한 원소씩 정수 타입으로 저장되어 있다.

- (d) 한편, 세 번째 파일인 `txt` 타입의 결과 값 출력 파일에는 계산 수행 결과를 기술하는 다섯 개의 정수 값들이 한 줄에 한 개씩 ASCII format으로 저장되어야 한다.

$s$   
 $k$   
 $i$   
 $l$   
 $j$

여기서  $s$ 는 여러분이 찾은 maximum subarray의 sum이고,  $k$ 와  $i$ 는 그 subarray의 위-왼쪽 모서리의 인덱스, 그리고  $l$ 와  $j$ 는 아래-오른쪽 모서리의 인덱스를 나타낸다(강의자료 1의 61쪽 그림 참조). 입력 데이터의 인덱스는 각 방향 0부터 시작하며, 만약 동일한 최대 합을 가지는 subarray이 여러 개가 있을 경우, 어떤 것에 대한 정보를 출력해도 무방하나, 가급적  $i \rightarrow j \rightarrow k \rightarrow l$  순서대로 인덱스가 가장 작은 subarray에 대한 인덱스를 출력하도록 하라.

### 채점 내용

1. 보고서의 가장 첫 부분에 자신이  $O(mn^2)$ 의 시간 복잡도를 가지는 알고리즘을 올바르게 구현하였는지 명확히 밝힐 것.
2. 여러분의 프로그램은 원하는 maximum sum subarray를 정확히 찾아주어야 한다. 채점은 제공한 데이터 외에 다른 입력 데이터를 사용하여 여러분의 프로그램이 정확한 결과를 산출하는지 **기계적으로 확인할 예정임. 따라서 입출력 파일의 형식 부분에 문제가 있을 경우 0점 처리가 됨.**
3. 주어진 크기에 대한 수행 시간이 보편적인 수행 시간보다 눈에 띄게 느릴 경우 0점 처리가 될 수 있음.
4. 다음과 같이 실제 수행 시간에 대한 실험을 진행한 후, 그 결과를 보고서에 기술하라.
  - (a) 수행 시간 및 시간 복잡도의 관계를 분석하기 위하여, 주어진 입력 크기  $(m, n)$ 에 대해 실험을 진행하라. 시간 측정은 **x64 플랫폼에서 Release 모드로** 컴파일하여 ms 단위로 측정하라. 특히 실험 결과의 신뢰도를 높이기 위하여 가급적 동일한 입력 영상에 대하여 여러 번 수행한 후 평균 시간을 취할 것.
  - (b) 이를 통하여 산출한 실험 결과로부터 자신이 구현한 방법의 이론적인 시간 복잡도와 수행 시간 간의 관계를 분석하고, 그로부터 자신이 발견한 사실과 그에 대한 의견을 보고서 파일에 **명확히** 기술하라.
    - 본 숙제의 중요한 목적 중의 하나는 이론적인 시간 복잡도와 실제 프로그램의 수행 시간과의 연관성을 찾아내는 것인데, 실험 결과를 **적절한 형식의 테이블로 요약**하라. 실제 측정한 시간 값과 이론적으로 구한 시간 복잡도의 연관성을 증명하기 위하여 그래프를 그린다던가 또는 수식으로 함수 관계를 보인다던가 하는 등의 **공학적인 방법**을 사용하여 자신의 주장을 **논리적으로** 기술할 것. 특히,  $(m, n)$ 과  $(n, m)$ 인 경우의 수행 시간에 차이가 있을지 확인할 것.
    - 보고서에 실험 결과를 기술하기 전에 본인이 사용한 컴퓨터의 실험 환경을 정확히 기술할 것 (아래의 예를 참조할 것).

OS: Windows 11 Education  
 CPU: Intel(R) Core(TM) i7-7700K CPU @ 4.20GHz  
 RAM: 16.00GB  
 Compiler: Visual Studio 22 Release Mode/x64 Platform

### 주의 사항

1. 숙제 제출 방법에 대하여 조교가 사이버 캠퍼스에 공지하는 내용을 숙지할 것.
2. 채점 시 형식 상의 문제 (예를 들어, configuration 파일의 내용을 정확히 읽어들이지 못할 경우) 테스트 데이터에 대하여 답을 찾지 못한 것으로 간주할 예정임. 다수의 학생에 대하여 기계적으로 채점을 해야하니 양해 바람.
3. 시간 측정은 본 수업에서 배포하는 예제 프로그램에서 사용한 시간 측정 방법을 이해하여 사용할 것.
4. 제출한 원시 코드에서 대해서는 copy-check를 수행할 예정이며, 다른 사람의 코드 또는 보고서를 복사할 경우 **관련된 사람 모두에 대하여 (즉 복사한 사람과 복사 당한 사람 모두) 과목 최종 성적의 50%를 감점함.**