

# 과제 #3

M3239.002300\_003 데이터 사이언스 특강

Due: 2021년 10월 20일 23시 59분

본 과제의 목표는 sorting 및 search 알고리즘의 big O time complexity와 실제 실행시간 간의 관계를 파악하는 것이다. 본 과제에서 처리해야 하는 데이터의 범위는 [0, 999]이다. 이러한 데이터를 원소로 가진 리스트가 입력으로 주어진다. 실험 수행 시 정확한 시간 측정을 위하여 다른 프로그램은 실행하지 않거나 혹은 실습 서버를 사용할 것을 권장한다.

## 1 문제 1: 정렬 구현 및 성능 분석 보고서 작성

수업 시간에 배운 정렬 알고리즘 중 bubble sort, insertion sort, selection sort, merge sort, quick sort(non-in-place), counting sort, radix sort 를 python으로 구현하라. 그 후 입력 리스트의 크기를 1, 10, 100, 1000, 10000, 100000으로 주어서 각 sorting 알고리즘을 실행하고, 실행시간을 측정하여 실행시간을 비교할 수 있는 하나의 그래프를 도시하라. 그래프의 X 축은 입력의 크기, Y 축은 실행시간(micro-seconds)을 밑이 10인 log 스케일로 나타낸다. Web 검색 등을 통해 Python time module에 있는 perf\_counter()의 사용법을 숙지하고, 실행 시간 측정 시 perf\_counter()를 이용한다. 아래에 대하여 실행시간을 비교할 수 있는 각각 하나의 그래프가 나와야 한다.

A. 각 algorithm의 best case를 고려하였을 때

B. 각 algorithm의 average case를 고려하였을 때

- 이 경우 입력은 random number 생성으로 만들되, 각각 다른 입력으로 5번 실행하여 평균을 취한다.
- Web 검색 등을 통해 random number 생성 방법을 숙지하라.

C. 각 algorithm의 worst case를 고려하였을 때

도시한 sorting algorithm 들의 실행시간에 대한 그래프를 보고 sorting 알고리즘의 big O time complexity와 실제 실행시간 간의 관계를 논하라. 또, 어떤 식으로 best case, average case, worst case를 고려하였는지 설명하라.

구현 시 유의할 사항은 아래와 같다.

- 구현 시 (1) 인자로 들어온 list object를 직접 수정하여 값을 정렬한 후 이를 반환하여도 되고, (2) 인자로 들어온 list object를 복제해서 새 object를 만든 후 이를 정렬하여 반환하여도 된다. 함수 실행 뒤 반환된 list가 올바르게 정렬된 값을 담고 있기만 하면 된다. 인자로 들어온 list object의 상태는 어떻게 되어도 상관 없다.
- 구현 시 강의 슬라이드에 있는 코드를 사용하여도 된다. 단, 제출 시 print 함수는 전부 삭제하고 제출할 것.

## 2 문제 2: Search 구현 및 성능 분석 보고서 작성

수업 시간에 배운 linear search 및 binary search를 python으로 구현하라. 그 후 입력 리스트의 크기를 1, 10, 100, 1000, 10000, 100000으로 주어서 각 search 알고리즘을 실행하고 실행시간을 측정하여 실행시간을 비교할 수 있는 하나의 그래프를 도시하라. 그래프의 X 축은 입력의 크기, Y 축은 실행시간(micro-seconds)을 밑이 10인 log 스케일로 나타낸다. 입력은 random number를 생성하여 만들고 각각 다른 입력으로 5번 실행하여 실행시간의 평균을 취한다. 아래에 대하여 실행시간을 비교할 수 있는 각각 하나의 그래프가 나와야 한다.

A. random number로 생성된 입력이 이미 sorting이 된 경우

B. random number로 생성된 입력이 sorting이 되지 않았을 경우

도시한 search algorithm 들의 실행시간에 대한 그래프를 보고 search 알고리즘의 big O time complexity와 실제 실행시간 간의 관계를 논하라.

### 3 구현 및 제출

실습 서버의 실습 계정 홈 디렉토리의 HW3 디렉토리에 prob1.py, prob2.pdf를 저장한다. 각각의 파일에는 아래와 같은 내용이 구현되어 있어야 한다.

파일	설명
prob1.py	bubble_sort, insertion_sort, selection_sort, merge_sort, quick_sort, counting_sort, radix_sort function 구현
prob1.pdf	성능 분석 보고서
prob2.py	linear_search, binary_search function 구현
prob2.pdf	성능 분석 보고서

제출 시 아래와 같은 사항을 유의하라.

- prob1.py 및 prob2.py 제출 파일에는 반드시 함수를 정의하는 코드만 포함되어야 하며, 테스트 코드 등이 포함되어서는 안 된다. 제출한 코드에 불필요한 내용이 포함되어 채점이 방해될 경우 해당 과제가 0점 처리될 수 있다.
- prob1.py 및 prob2.py 파일에서는 별도의 module을 import하지 않고 python 기본 기능만으로 함수들을 구현하도록 한다. prob1.pdf, prob2.pdf 보고서를 작성하기 위한 실험 수행 시 필요한 module들을 import하여 사용하여도 된다(random, time 등). 실험을 위해 사용한 코드는 별도로 제출하지 않아도 된다.
- 보고서에 학번, 실습 서버 id (예: spds000), 이름을 명기할 것
- 보고서의 분량은 자유이다. 필요한 내용이 전부 들어있다면 총 분량이 적어도 만점을 받을 수 있다. 흥미로운 추가 실험 및 분석이 포함될 경우 추가 점수를 받을 수 있다. 흥미로운 내용 없이 분량이 불필요하게 길 경우 조교가 힘들어 할 수 있다.
- 실험 환경은 자유이다. 실습용 서버에서 성능을 측정해도 되고, 각자의 PC에서 성능을 측정해도 된다. 단, 실습용 서버를 사용하지 않는 경우 실험 환경에 대한 간단한 요약이 보고서에 포함되어야 한다 (Python 버전, CPU 종류, 메모리 크기 등). 실습용 서버에서 성능을 측정할 경우, 제출 마감 직전에는 많은 유저가 동시에 프로그램을 실행하기 때문에 실험 결과가 영향을 받을 수 있다. 이에 유의하여 실험을 진행하여야 한다.
- 보고서는 반드시 **prob1.pdf, prob2.pdf** 라는 이름의 pdf 포맷의 파일로 제출하여야 한다. Microsoft Word, 한컴오피스, Latex 등 대부분의 문서 작성 프로그램에서 pdf로 문서를 export하는 기능을 지원한다. 방법을 모르겠는 경우 조교에게 메일로 문의할 것.
- 실습 서버의 HW3 디렉토리에 파일을 저장하면, 과제 제출 시간에 자동으로 해당 파일을 복사해 간다. 파일명을 다른 이름으로 수정할 경우 (예: p1.py) 과제가 제출되지 않을 수 있으니 유의하여야 한다.
- 그레이스 데이를 사용할 경우 spdsta@aces.snu.ac.kr로 이를 알려야 하며, 메일 제목은 [SPDS] HW3-Graceday 학번 이름의 포맷으로 작성하라 (예: [SPDS] HW3-Graceday 2020-20000 안규수).