

# 2022-1 Algorithm Homework1 보고서

서어서문학과 2017-16849 이윤경

CLRS의 선형 선택 알고리즘(Randomized Select)과 최악의 경우에도 선형의 시간을 보장하는 선택 알고리즘(Deterministic-select) 알고리즘을 파이썬으로 구현하였다. Randomized Select 알고리즘에서 Randomized-partition의 random pivot을 구현하기 위해 random 라이브러리의 random.randint()를 사용하였다. 또한, 알고리즘의 시간 측정을 위해 time 라이브러리를 사용하였다.

## 1. I/O 및 checker program

파일의 입력은 첫 번째 줄에 원소의 개수인  $n$ , 두 번째 줄에  $n$ 개의 원소를 공백으로 구분하여 입력하였고, 마지막 줄에 프로그램이 찾을 원소의 index인  $i$ 로 구성되어 있다. 이 input의 각 줄을 읽어  $n$ 과  $r\_numbers$ ,  $d\_numbers$ ,  $c\_numbers$ ,  $i$ 의 변수에 저장하였다. numbers가 정렬되는 과정에서 값이 달라지는 것을 막기 위해 각 함수에서 사용될 numbers의 값을 사전에 저장해 두었다. 출력은 deter.txt와 random.txt에 각각  $i$ 번째 값과 알고리즘의 실행 시간을 ms단위로 출력하였다.

checker program은 numbers,  $i$ , output을 입력 받아 전체 원소를 한 번씩 훑으며 deter.txt와 random.txt의 결과값인 output보다 작은 값이 있으면 count 변수에 1을 더하고, 같은 값이 있으면 equal 변수에 1을 더하여 input의  $i$  값이 count 이상, count+equal-1 이하 일 때 "correct!" string을, 범위를 벗어났을 때는 "wrong!" string을 출력하는 방식으로 구현하였다. 각 원소를 한 번씩 지나갈 때  $n$ 번의 시간이 소요되는 것 이외에 다른 과정은 모두 상수 시간에 불과하므로 checker program은 선형 시간을 보장한다.

## 2. 실행 환경 및 구현 방식

vs code를 통해 코드를 구현하였고, Python 3.9.6 버전을 이용하였다. 윈도우 환경에서 테스트를 진행하였으며, hw1.py 파일 자체를 실행하거나 터미널에서 hw1.py가 존재하는 디렉토리로 이동 후, python hw1.py를 입력하여 실행 가능하다. input.txt는 hw1.py와 같은 디렉토리에 존재하여야 한다.

Random-Selection은 CLRS의 pseudo code를 참고하여 구현하였으며, 임의의 pivot을 위해 randint()를 사용하였다. Deterministic-Selection에서 삽입정렬은 쉽게 배우는 알고리즘의 pseudo code를 참고하여 구현하였다. 또한 median을 찾을 때, 먼저  $n$ 을 5로 나눈 몫으로 각 그룹을 구분하여 삽입정렬로 정렬하여 중간값을 구한 후,  $n$ 이 5의 배수가 아닐 경우 마지막 그룹의 중간값도 동일한 방식으로 구하였다. 이러한 중간값들을 medians list에 삽입한 후, 중앙값들의 중앙값 medianum을 구하기 위하여 dselect 함수를 재귀 호출하였다. 이후 partition이나  $i$ 번째 값을 찾는 과정은 Random-Selection과 유사하게 구현하였다.

sample case의 input은  $n$ 개의 1보다 크거나 같고 1,000,000보다 작거나 같은 임의의 정수로 구성되어 있으며,  $i$  또한 1 보다 크거나 같고  $n$ 보다 작거나 같은 정수이다. 각각의 sample case 폴더에 input.txt, random.txt, deter.txt, result.txt가 저장되어 있으며 sample case에 있는 hw1.py 파일은 모두

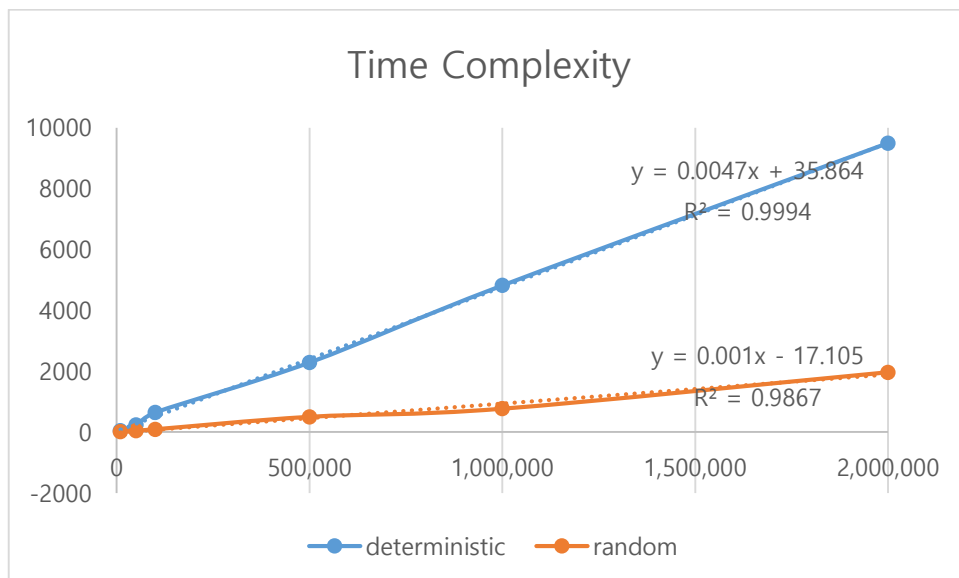
동일한 파일로, hw1.py 파일을 통해 프로그램을 실행할 수 있다.

### 3. 실행 결과 및 example running

n의 크기에 따른 알고리즘 실행 시간은 다음 표와 같다. 표의 가로축은 입력의 크기인 n이며, 각 크기 별로 4개의 test case를 통해 실행 시간을 측정하였다. 그리고 n에 따른 시간의 평균을 기록하여 그래프로 나타내었다.

random	10,000	50,000	100,000	500,000	1,000,000	2,000,000
case 1	16	43	109	318	378	3747
case 2	16	31	98	708	640	1378
case 3	16	62	81	428	629	1829
case 4	31	47	78	550	1423	909
average	19.75	45.75	91.5	501	767.5	1965.75

deterministic	10,000	50,000	100,000	500,000	1,000,000	2,000,000
case 1	47	362	634	2502	4069	11954
case 2	52	207	702	2250	3923	8298
case 3	31	190	602	2199	4277	9674
case 4	56	175	658	2203	7002	8047
average	46.5	233.5	649	2288.5	4817.75	9493.25

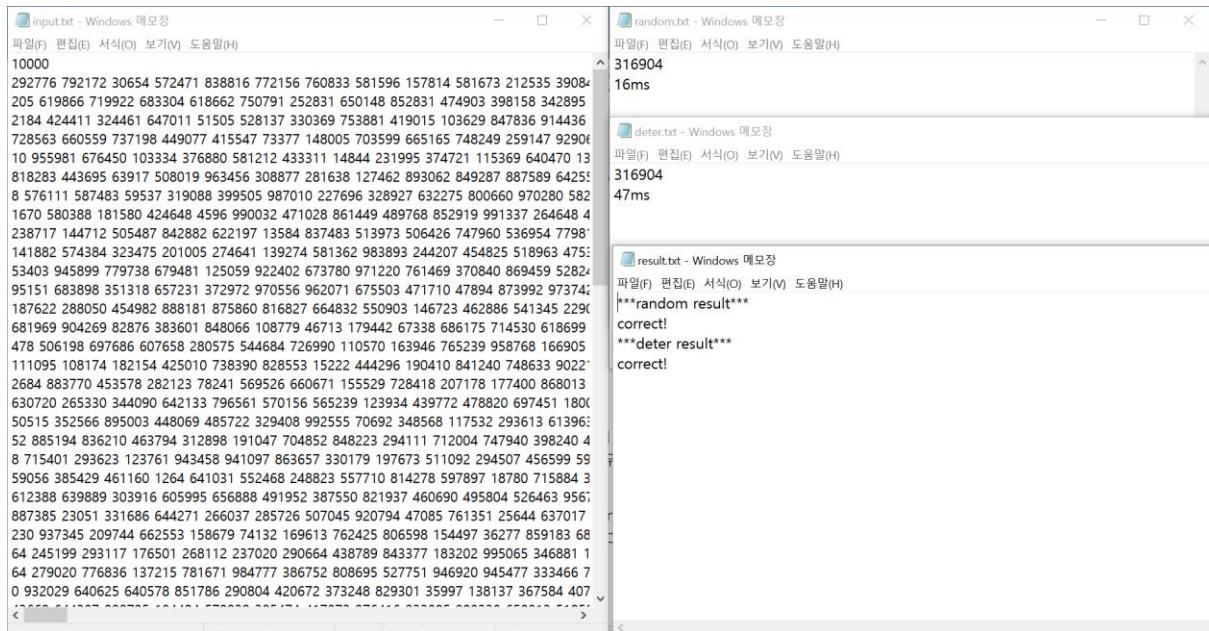


그 결과 두 그래프 모두 추세선의  $R^2$ 값이 1에 가까워 시간복잡도가  $\theta(n)$ 임을 확인할 수 있다. 두 추세선의 관계식은 Deterministic-selection의 경우  $y = y = 0.0047x + 35.864$ , Random-selection의 경우  $y = 0.001x - 17.105$ 로, 기울기의 비율은 4.7이다. 두 알고리즘의 시간 비율은 4.7로 Deterministic-selection 알고리즘이 Random-selection보다 더 오래 걸린다는 것을 알 수 있다. 그러나 이는 평균적인

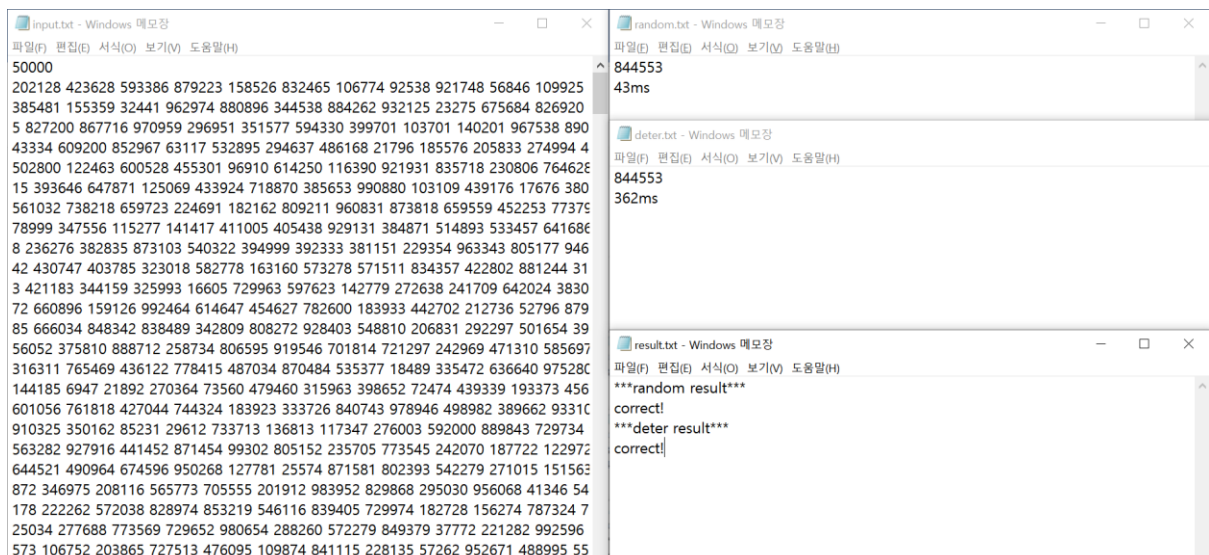
값으로 원소의 구성이나 찾고자 하는 i번째 원소의 위치에 따라 달라질 수 있다. 또한 Random-selection은 pivot을 임의의 수로 지정하기 때문에 경우에 따라 시간의 편차가 상대적으로 크게 나타날 수 있다.

각 case의 시간 복잡도 이외의 example running 결과는 문서로 담는 데에 용량의 한계로 인해 첨부된 sample 폴더의 각 txt 파일로 대신한다. 그 중 일부의 결과는 다음과 같다.

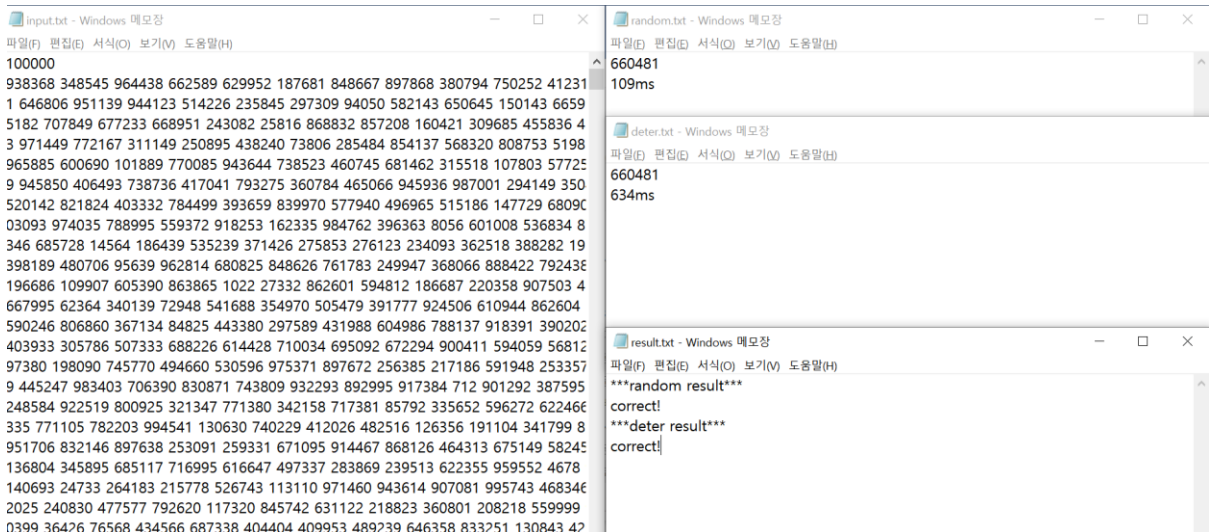
### 1) n=10,000 (sample1\_1)



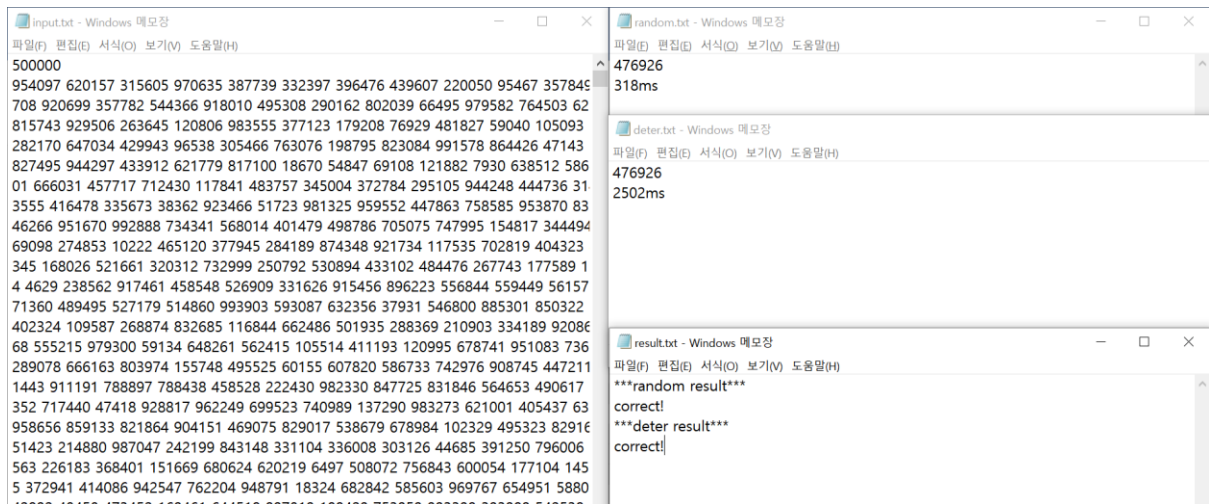
### 2) n=50,000 (sample2\_1)



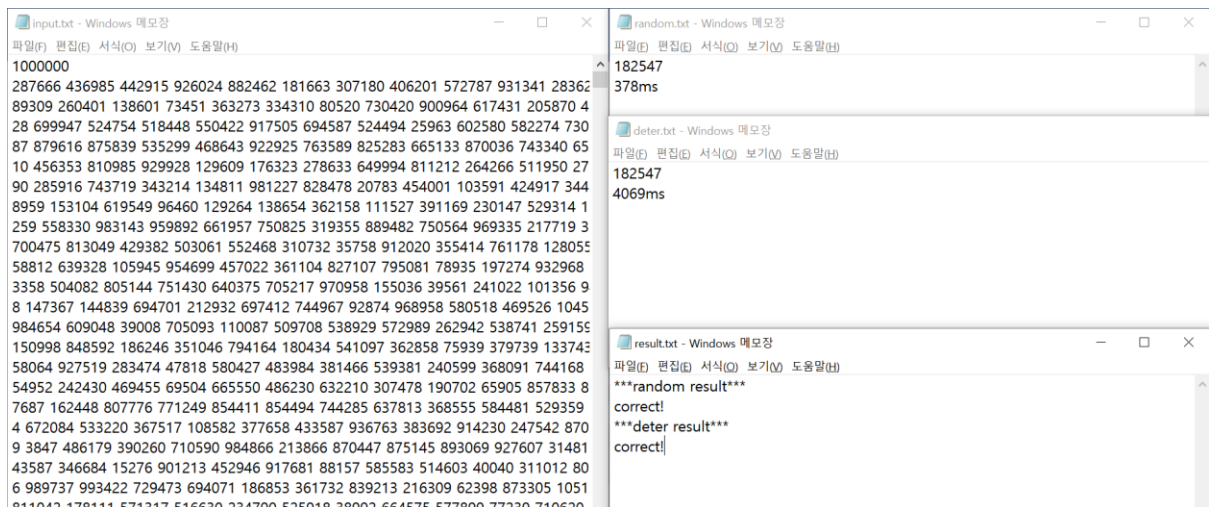
### 3) n=100,000 (sample3\_1)



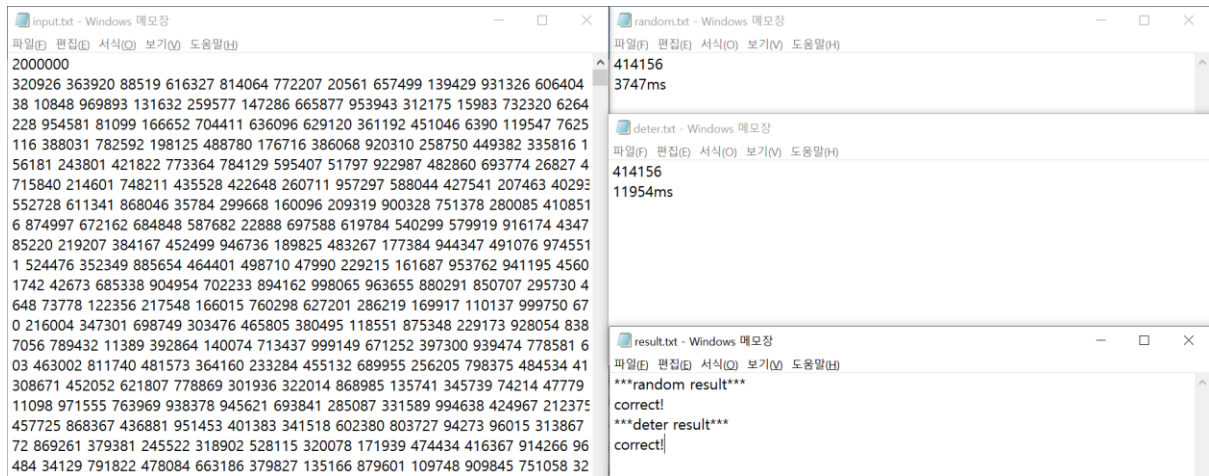
4) n=500,000 (sample4\_1)



5) n=1,000,000 (sample5\_1)



6) n=2,000,000 (sample6\_1)



The image shows three overlapping Notepad++ windows. The 'input.txt' window on the left contains a large list of 2,000,000 integers. The 'random.txt' window on the top right shows two values: 414156 and 3747ms. The 'result.txt' window on the bottom right shows the output of a comparison, indicating that both 'random' and 'deter' results are correct.

```
input.txt - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
2000000
320926 363920 88519 616327 814064 772207 20561 657499 139429 931326 606404
38 10848 969893 131632 259577 147286 665877 953943 312175 15983 732320 6264
228 954581 81099 166652 704411 636096 629120 361192 451046 6390 119547 7625
116 388031 782592 198125 488780 176716 386068 920310 258750 449382 335816 1
56181 243801 421822 773364 784129 595407 51797 922987 482860 693774 26827 4
715840 214601 748211 435528 422648 260711 957297 588044 427541 207463 40293
552728 611341 868046 35784 299668 160096 209319 900328 751378 280085 410851
6 874997 672162 684848 587682 22888 697588 619784 540299 579919 916174 4347
85220 219207 384167 452499 946736 189825 483267 177384 944347 491076 974551
1 524476 352349 885654 464401 498710 47990 229215 161687 953762 941195 4560
1742 42673 685338 904954 702233 894162 998065 963655 880291 850707 295730 4
648 73778 122356 217548 166015 760298 627201 286219 169917 110137 999750 67
0 216004 347301 698749 303476 465805 380495 118551 875348 229173 928054 838
7056 789432 11389 392864 140074 713437 999149 671252 397300 939474 778581 6
03 463002 811740 481573 364160 233284 455132 689955 256205 798375 484534 41
308671 452052 621807 778869 301936 322014 868985 135741 345739 74214 47779
11098 971555 763969 938378 945621 693841 285087 331589 994638 424967 212375
457725 868367 436881 951453 401383 341518 602380 803727 94273 96015 313867
72 869261 379381 245522 318902 528115 320078 171939 474434 416367 914266 96
484 34129 791822 478084 663186 379827 135166 879601 109748 909845 751058 32

random.txt - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
414156
3747ms

deter.txt - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
414156
11954ms

result.txt - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
***random result***
correct!
***deter result***
correct!
```