Algorithm HW #2

컴파일 환경: g++ (Ubuntu 7.4.0-1ubuntu1~18.04.1) 7.4.0

2-1. Hybrid Quick Sort with Insertion Sort * Median of 3 Partitioning

※ 파일 이름: 2-1.cpp, 2-1.out

1) 헤더 파일 목록

- <iostream>, <fstream>, <cstdlib>, <ctime>, <stdlib.h>

2) Functions

function	설명
void InsertionSort(int* array, int p, int r);	시작 index가 p, 마지막 index가 r인 array를 받아 insertion sort를 수행한다.
void HybridQuickSort(int* array, int p, int r);	시작 index가 p, 마지막 index가 r인 array를 받아 Hybrid Quick sort를 수행한다. Random하게 세 값을 골라 그 값 중 가운데 값을 pivot으로 하고, 원소 개수가 10개 이하일 때는 insertion sort를 수행한다.
int Partition(int* array, int p, int r);	pivot 값을 기준으로 시작 index가 p, 마지막 index가 r인 array를 pivot보다 큰 값과 작은 값 그룹으로 나눈다. pivot값은 array[r]에 저장되어 있다.(이 함수를 실행시키기 전에 pivot을 정한 뒤 array[pivot]을 array[r]과 바꾸기 때문이다.) 나눈 뒤 pivot의 index를 return한다.
void Swap(int* a, int* b);	a, b의 값을 서로 바꿔준다.
int Mid(int* array, int a, int b, int c);	array[a], array[b], array[c] 값 중 중간 값의 index를 return한다.
int Rand(int p, int r)	p와 r 사이에 있는 값을 랜덤으로 return한다.

3) main 함수 동작 과정

① 배열 A에 200,000만 개의 자리를 할당한 뒤 input2-1.txt 파일이 끝날 때까지 원소를 받고 원소 개수에 맞춰 배열 A의 크기를 재할당한다.

② HybridQuickSort를 실행시키고 실행 전후의 clock을 기록한다.

③ 정렬된 배열 A와 정렬하는데 걸린 시간을 output2-1.txt로 출력한다.

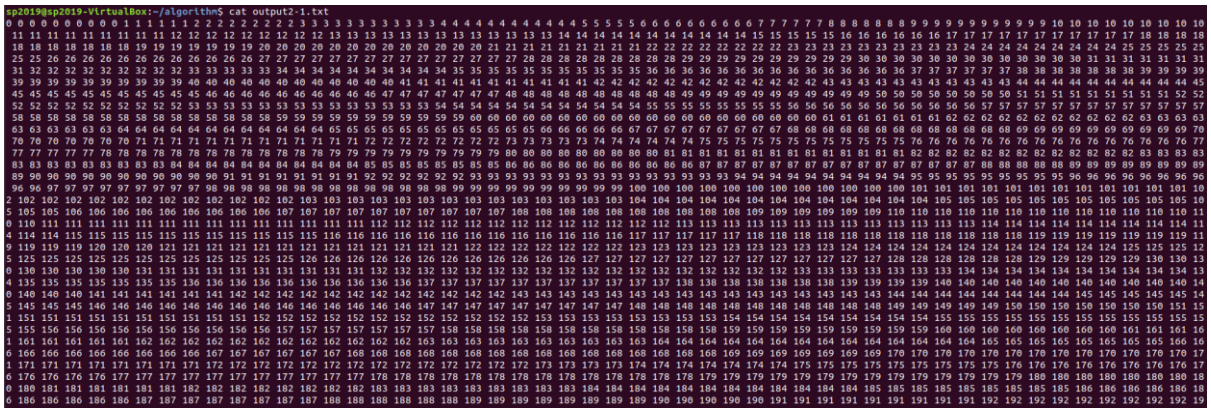
④ A의 동적할당을 해제하고 ifstream과 ofstream을 close한다.

4) 실행 화면 캡처

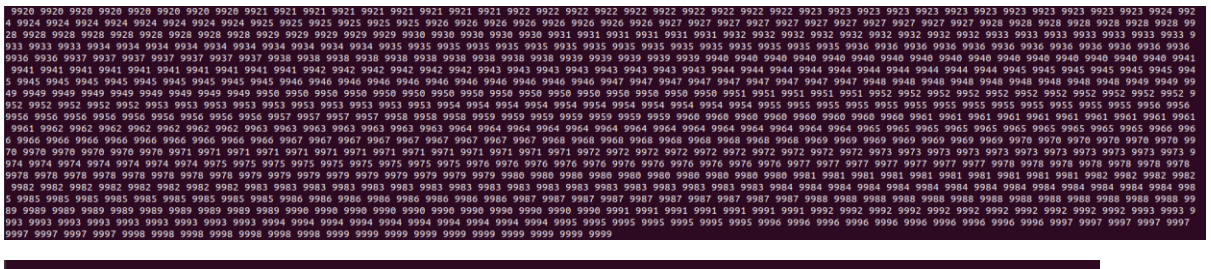
- 주어진 예시

```
sp2019@sp2019-VirtualBox:~/algorithm$ cat input2-1.txt
5 3 99 8 5 39 111 989 31 88 87 99 4 187 553 2
sp2019@sp2019-VirtualBox:~/algorithm$ ./2-1.out
sp2019@sp2019-VirtualBox:~/algorithm$ cat output2-1.txt
2 3 4 5 5 8 31 39 87 88 99 99 111 187 553 989
0.000003 s
```

- portal에 올라온 예시



(중간생략)



2-2. Rod Cutting

※ 파일 이름: 2-2.cpp, 2-2.out

1) 헤더 파일 목록

- <iostream>, <fstream>, <utility>

2) Functions

function	설명
pair<int*, int*> ExtendedRodCut(int* p, int n)	각 크기의 가격이 저장된 배열 p에서 길이 n짜리 통나무를 Bottom-Up-Cut한다. 배열 r에는 각 길이 별 받을 수 있는 최대 가격을, s에는 쪼갠 위치를 저장한다. Pair <r, s>를 return한다.
void Print(int* p, int n)	ExtendedRodCut를 실행시킨 후 길이 n에서의 최대 길이를 첫 줄에, 어떻게 잘랐는지를 둘째 줄에 출력한다.

3) main 함수 동작 과정

① 배열 p를 동적할당한 뒤 input2-1.txt 파일을 통해 통나무의 길이와 각 길이 별 가격을 입력 받는다.

② Print 함수를 통해 ExtendedRodCut 함수로 최대 가격을 계산하고 출력한다.

③ 배열 p의 동적할당을 해제하고 ifstream을 close한다.

4) 실행 화면 캡처

- 주어진 예시

```
sp2019@sp2019-VirtualBox:~/algorithm$ cat input2-2.txt
5
1 5 8 9 10
sp2019@sp2019-VirtualBox:~/algorithm$ ./2-2.out
13
2 3
```

- 강의자료 예시

```
sp2019@sp2019-VirtualBox:~/algorithm$ cat input2-2.txt
10
1 5 8 9 10 17 17 20 24 30
sp2019@sp2019-VirtualBox:~/algorithm$ ./2-2.out
30
10
```

- 추가 예시

```
sp2019@sp2019-VirtualBox:~/algorithm$ cat input2-2.txt
20
2 4 7 9 10 15 17 20 22 24 28 30 34 35 35 39 41 46 48 51
sp2019@sp2019-VirtualBox:~/algorithm$ ./2-2.out
51
1 6 13
```