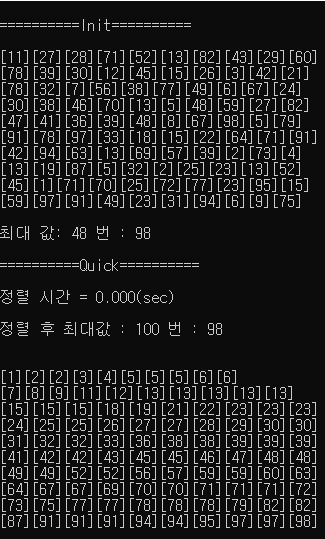
|  |  |  |  |
| --- | --- | --- | --- |
| 2022/2 『자료구조』실습 보고서 | | | |
| 제목 | 1장 실습( O ) 과제( ) | 제출일자 | 2022.  09 .    08. |
| 학번 | 201911608 | 이름 | 김지환 |

code

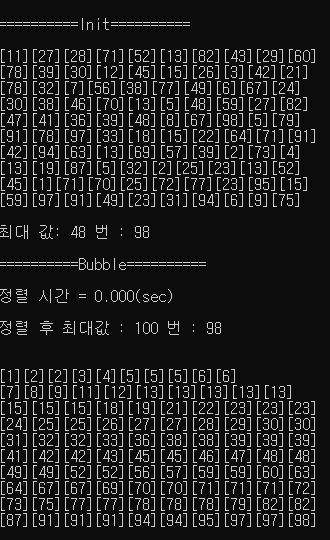
|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <time.h>  #include <algorithm>  #define MAX 100  int random[MAX];  void rand\_Number(){  srand((unsigned int)time(NULL));  for (int i = 0; i < MAX; i++) {  random[i] = rand()%100;  if (i%10==0) printf("\n");  printf("[%d]", random[i]);  }  }  void max\_Number() {  int max = 0;  int idx = 0;  for (int i =0; i<MAX; i++)  if (max < random[i]){  max = random[i];  idx = i+1;  }  printf("%d 번 : %d\n\n", idx, max);  }  void init(void) {  printf("\n==========Init==========\n");  rand\_Number();  printf("\n\n최대 값: ");  max\_Number();  }  void swap(int a, int b) {  int temp = random[a];  random[a] = random[b];  random[b] = temp;  }  void selection\_Sort() {  printf("==========Selection==========\n");  for(int i =0;i<MAX-1;i++)  for(int j=i+1;j<MAX;j++)  if(random[i]>random[j]) swap(i, j);  }  void bubble\_Sort() {  printf("==========Bubble==========\n");  for (int i = 0; i < MAX; i++)  for (int j = 0; j < MAX-i-1; j++)  if (random[j] > random[j + 1]) swap(j, j+1);  }  void insert\_Sort() {  printf("==========Insert==========\n");  for (int i = 0; i < MAX-1; i++) {  int j = i;  while (random[j] > random[j + 1]) {  swap(j, j+1);  j--;  }  }  }  void quick\_Sort(int start, int end) {  if(start >= end) return;  int key = start;  int i = start + 1;  int j = end;  int temp;  while (i <= j) {  while (random[i] <= random[key]) i++;  while (random[j] >= random[key] && j > start) j--;  if (i > j) swap(j, key);  else swap(j, i);  }  quick\_Sort(start, j - 1);  quick\_Sort(j + 1, end);  }  int main() {    init();  printf("==========Quick==========\n");  int start = clock();  quick\_Sort(0,MAX-1);  printf("\n정렬 시간 = %.3lf(sec)\n", (double)(clock() - start) / CLOCKS\_PER\_SEC);  printf("\n정렬 후 최대값 : ");  max\_Number();  for (int i=0; i<MAX; i++) {  if (i%10==0) printf("\n");  printf("[%d]", random[i]);  }  printf("\n");    init();  start = clock();  bubble\_Sort();  printf("\n정렬 시간 = %.3lf(sec)\n", (double)(clock() - start) / CLOCKS\_PER\_SEC);  printf("\n정렬 후 최대값 : ");  max\_Number();  for (int i=0; i<MAX; i++) {  if (i%10==0) printf("\n");  printf("[%d]", random[i]);  }  printf("\n");    init();  start = clock();  insert\_Sort();  printf("\n정렬 시간 = %.3lf(sec)\n", (double)(clock() - start) / CLOCKS\_PER\_SEC);  printf("\n정렬 후 최대값 : ");  max\_Number();  for (int i=0; i<MAX; i++) {  if (i%10==0) printf("\n");  printf("[%d]", random[i]);  }  printf("\n");    init();  start = clock();  selection\_Sort();  printf("\n정렬 시간 = %.3lf(sec)\n", (double)(clock() - start) / CLOCKS\_PER\_SEC);  printf("\n정렬 후 최대값 : ");  max\_Number();  for (int i=0; i<MAX; i++) {  if (i%10==0) printf("\n");  printf("[%d]", random[i]);  }  printf("\n");    init();  start = clock();  printf("==========Library==========\n");  std::sort(random,random+MAX);  printf("\n정렬 시간 = %.3lf(sec)\n", (double)(clock() - start) / CLOCKS\_PER\_SEC);  printf("\n정렬 후 최대값 : ");  max\_Number();  for (int i=0; i<MAX; i++) {  if (i%10==0) printf("\n");  printf("[%d]", random[i]);  }  printf("\n");  return 0;  } |

결과

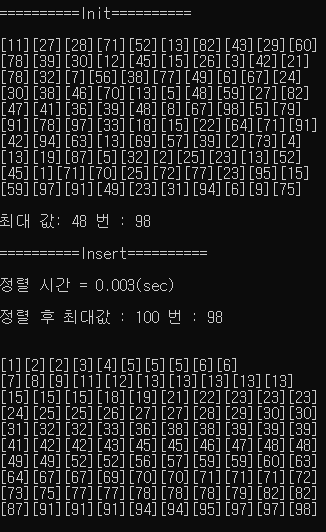
QUICK 정렬



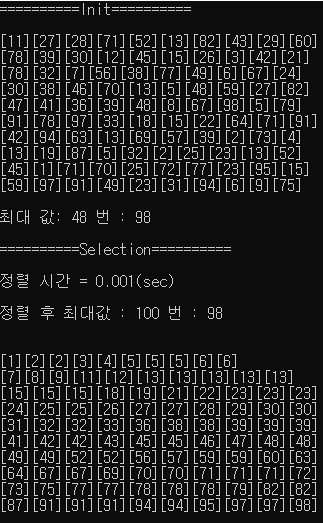
Bubble



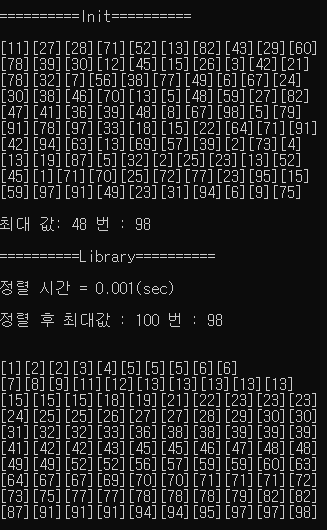
Insert



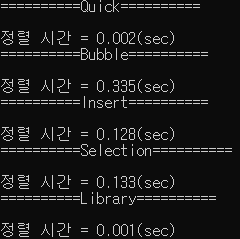
Selection



algorithm Library



data를 10000개로 수정 후 결과 값:



data가 클 수록 O(log2 N)의 Quick 정렬이 가장 빠름을 알 수 있고

Library가 가장 빠르다.