알고리즘
Homework_2

과목명	알고리즘_01
담당 교수	주종화
학 과	컴퓨터공학과
제출일	2024.09.29
학 번	2021111945
성 명	송하연

문제 1. 버블 정렬 (50 점)

1. 비순환적 버블정렬을 구현해 보아라. (25 점, 부분 점수 없음)

□ 유사코드를작성하시오

```
Function Bubble(A,n):
    Sorted = false
    while Not Sorted:
        Sorted = true
        for i From 0 To n-2
            if A[i] > A[i+1] Then
                swap A[i]와 A[i+1]
                Sorted = false
        Print Array(A)
```

□ 프로그램을 작성하고 입력 A=[30 20 40 10 5 10 30 15]에 대한 정렬 결과를 출력하여라. 단계별로 A 값이 어떻게 변화되는지 (뭔가 배열에 값이 바뀌거나 각단계마다 A 를 출력하면 됨) 와 최종 결과물을 출력할것.

```
========== 2021111945 송하연 ==========
배열의 크기를 입력하세요 : 8
배열의 요소를 입력하세요(8개 입력) : 30 20 40 10 5 10 30 15
정렬 방법을 선택하세요 (1: 비순환적, 2: 순환적) : 1
========== 비순환적 버블 정렬 ==========
1단계 : [20] [30] [40] [10] [5] [10] [30] [15]
1단계 : [20] [30] [40] [10] [5] [10] [30] [15]
1단계 : [20] [30] [10] [40] [5] [10] [30] [15]
1단계 : [20] [30] [10] [5] [40] [10] [30] [15]
1단계 : [20] [30] [10] [5] [10] [40] [30] [15]
1단계 : [20] [30] [10] [5] [10] [30] [40] [15]
1단계 : [20] [30] [10] [5] [10] [30] [15] [40]
2단계 : [20] [30] [10] [5] [10] [30] [15] [40]
2단계 : [20] [10] [30] [5] [10] [30] [15] [40]
2단계 : [20] [10] [5] [30] [10] [30] [15] [40]
2단계 : [20] [10] [5] [10] [30] [30] [15] [40]
2단계 : [20] [10] [5] [10] [30] [30] [15] [40]
2단계 : [20] [10] [5] [10] [30] [15] [30] [40]
2단계 : [20] [10] [5] [10] [30] [15] [30] [40]
3단계 : [10] [20] [5] [10] [30] [15] [30] [40]
3단계 : [10] [5] [20] [10] [30] [15] [30] [40]
3단계 : [10] [5] [10] [20] [30] [15] [30] [40]
3단계 : [10] [5] [10] [20] [30] [15] [30] [40]
3단계 : [10] [5] [10] [20] [15] [30] [30] [40]
3단계 : [10] [5] [10] [20] [15] [30] [30] [40]
3단계 : [10] [5] [10] [20] [15] [30] [30] [40]
4단계 : [5] [10] [10] [20] [15] [30] [30] [40]
4단계 : [5] [10] [10] [20] [15] [30] [30] [40]
4단계 : [5] [10] [10] [20] [15] [30] [30] [40]
4단계 : [5] [10] [10] [15] [20] [30] [30] [40]
4단계 : [5] [10] [10] [15] [20] [30] [30] [40]
4단계 : [5] [10] [10] [15] [20] [30] [30] [40]
4단계 : [5] [10] [10] [15] [20] [30] [30] [40]
5단계 : [5] [10] [10] [15] [20] [30] [30] [40]
5단계 : [5] [10] [10] [15] [20] [30] [30] [40]
5단계 : [5] [10] [10] [15] [20] [30] [30] [40]
5단계 : [5] [10] [10] [15] [20] [30] [30] [40]
5단계 : [5] [10] [10] [15] [20] [30] [30] [40]
5단계 : [5] [10] [10] [15] [20] [30] [30] [40]
```

2. 이를 순환적 버블정렬로 구현해 보아라. (25 점, 부분 점수 없음)

□ 유사코드를작성하시오

```
Function RecursiveBubble(A,n):
    if n<=1 Then Retun
    for i From 0 To n-2
        if A[i] > A[i+1] Then
            swAp A[i]와 A[i+1]
    PrintArray(A)

    CALL RecursiveBubble(A,n-1)
```

□ 프로그램을 작성하고 입력 A=[30 20 40 10 5 10 30 15]에 대한 정렬 결과를 출력하여라. 단계별로 A 값이 어떻게 변화되는지(뭔가 배열에 값이 바뀌거나 각단계마다 A 를 출력하면 됨)와 최종 결과물을 출력할것

```
배열의 크기를 입력하세요 : 8
배열의 요소를 입력하세요(8개 입력) : 30 20 40 10 5 10 30 15
정렬 방법을 선택하세요 (1: 비순환적, 2: 순환적) : 2
========== 순환적 버블 정렬 ==========
1단계 : [20] [30] [40] [10] [5] [10] [30] [15]
1단계 : [20] [30] [40] [10] [5] [10] [30] [15]
1단계 : [20] [30] [10] [40] [5] [10] [30] [15]
1단계 : [20] [30] [10] [5] [40] [10] [30] [15]
1단계 : [20] [30] [10] [5] [10] [40] [30] [15]
1단계 : [20] [30] [10] [5] [10] [30] [40] [15]
1단계 : [20] [30] [10] [5] [10] [30] [15] [40]
2단계 : [20] [30] [10] [5] [10] [30] [15] [40]
2단계 : [20] [10] [30] [5] [10] [30] [15] [40]
2단계 : [20] [10] [5] [30] [10] [30] [15] [40]
2단계 : [20] [10] [5] [10] [30] [30] [15] [40]
2단계 : [20] [10] [5] [10] [30] [30] [15] [40]
2단계 : [20] [10] [5] [10] [30] [15] [30] [40]
2단계 : [20] [10] [5] [10] [30] [15] [30] [40]
3단계 : [10] [20] [5] [10] [30] [15] [30] [40]
3단계 : [10] [5] [20] [10] [30] [15] [30] [40]
3단계 : [10] [5] [10] [20] [30] [15] [30] [40]
3단계 : [10] [5] [10] [20] [30] [15] [30] [40]
3단계 : [10] [5] [10] [20] [15] [30] [30] [40]
3단계 : [10] [5] [10] [20] [15] [30] [30] [40]
3단계 : [10] [5] [10] [20] [15] [30] [30] [40]
4단계 : [5] [10] [10] [20] [15] [30] [30] [40]
4단계 : [5] [10] [10] [20] [15] [30] [30] [40]
4단계 : [5] [10] [10] [20] [15] [30] [30] [40]
4단계 : [5] [10] [10] [15] [20] [30] [30] [40]
4단계 : [5] [10] [10] [15] [20] [30] [30] [40]
4단계 : [5] [10] [10] [15] [20] [30] [30] [40]
4단계 : [5] [10] [10] [15] [20] [30] [30] [40]
5단계 : [5] [10] [10] [15] [20] [30] [30] [40]
5단계 : [5] [10] [10] [15] [20] [30] [30] [40]
5단계 : [5] [10] [10] [15] [20] [30] [30] [40]
5단계 : [5] [10] [10] [15] [20] [30] [30] [40]
5단계 : [5] [10] [10] [15] [20] [30] [30] [40]
5단계 : [5] [10] [10] [15] [20] [30] [30] [40]
5단계 : [5] [10] [10] [15] [20] [30] [30] [40]
```

문제 2. 퀵 정렬 (50 점)

□ 1~100000 까지 숫자 중 랜덤으로 10 개의 숫자를 갖는 배열 N 을 퀵 정렬을 사용하여 내림차순으로 정렬시키려 한다. (* 중복 없이)

□ 순환적 퀵 정렬을 구현해 보아라. (25 점)

□ 코드 제출

□ 정렬 결과를 출력하여라. 단계별로 값이 어떻게 변화되는지와 최종 결과물을 출력할것.

```
========== 2021111945 송하연 ==========


========== 정렬 전 배열  ==========
[7904] [22143] [35332] [37089] [37881] [41863] [50425] [50621] [72206] [91861]

========== 순환적 퀵 정렬 ==========
배열 상태: [91861] [22143] [35332] [37089] [37881] [41863] [50425] [50621] [72206] [7904]
선택된 피벗: [7904]  Left: 0  Right: 9

배열 상태: [91861] [22143] [35332] [37089] [37881] [41863] [50425] [50621] [72206]
선택된 피벗: [91861]  Left: 0  Right: 8

배열 상태: [91861] [72206] [35332] [37089] [37881] [41863] [50425] [50621] [22143]
선택된 피벗: [22143]  Left: 1  Right: 8

배열 상태: [91861] [72206] [35332] [37089] [37881] [41863] [50425] [50621]
선택된 피벗: [72206]  Left: 1  Right: 7

배열 상태: [91861] [72206] [50621] [37089] [37881] [41863] [50425] [35332]
선택된 피벗: [35332]  Left: 2  Right: 7

배열 상태: [91861] [72206] [50621] [37089] [37881] [41863] [50425]
선택된 피벗: [50621]  Left: 2  Right: 6

배열 상태: [91861] [72206] [50621] [50425] [37881] [41863] [37089]
선택된 피벗: [37089]  Left: 3  Right: 6

배열 상태: [91861] [72206] [50621] [50425] [37881] [41863]
선택된 피벗: [50425]  Left: 3  Right: 5

배열 상태: [91861] [72206] [50621] [50425] [41863] [37881]
선택된 피벗: [37881]  Left: 4  Right: 5


========== 순환적 퀵 정렬 최종 결과 ==========
[91861] [72206] [50621] [50425] [41863] [37881] [37089] [35332] [22143] [7904]
```

□ 비순환적 퀵 정렬로 구현해 보아라. (25 점)

□ 코드 제출

□ 정렬 결과를 출력하여라. 단계별로 값이 어떻게 변화되는지와 최종 결과물을 출력할것.

```
========== 정렬 전 배열 ==========
[7904] [22143] [35332] [37089] [37881] [41863] [50425] [50621] [72206] [91861]

========== 비순환적 퀵 정렬 ==========
배열 상태: [91861] [22143] [35332] [37089] [37881] [41863] [50425] [50621] [72206] [7904]
선택된 피벗: [7904]  Left: 0  Right: 9

배열 상태: [91861] [22143] [35332] [37089] [37881] [41863] [50425] [50621] [72206] [7904]
선택된 피벗: [91861]  Left: 0  Right: 8

배열 상태: [91861] [72206] [35332] [37089] [37881] [41863] [50425] [50621] [22143] [7904]
선택된 피벗: [22143]  Left: 1  Right: 8

배열 상태: [91861] [72206] [35332] [37089] [37881] [41863] [50425] [50621] [22143] [7904]
선택된 피벗: [72206]  Left: 1  Right: 7

배열 상태: [91861] [72206] [50621] [37089] [37881] [41863] [50425] [35332] [22143] [7904]
선택된 피벗: [35332]  Left: 2  Right: 7

배열 상태: [91861] [72206] [50621] [37089] [37881] [41863] [50425] [35332] [22143] [7904]
선택된 피벗: [50621]  Left: 2  Right: 6

배열 상태: [91861] [72206] [50621] [50425] [37881] [41863] [37089] [35332] [22143] [7904]
선택된 피벗: [37089]  Left: 3  Right: 6

배열 상태: [91861] [72206] [50621] [50425] [37881] [41863] [37089] [35332] [22143] [7904]
선택된 피벗: [50425]  Left: 3  Right: 5

배열 상태: [91861] [72206] [50621] [50425] [41863] [37881] [37089] [35332] [22143] [7904]
선택된 피벗: [37881]  Left: 4  Right: 5


========== 비순환적 퀵 정렬 최종 결과 ==========
[91861] [72206] [50621] [50425] [41863] [37881] [37089] [35332] [22143] [7904]
```

□ 위 두 경우 단계별로 값이 어떻게 변화되는지는 자신만의 방식으로 퀵정렬이 되어가고 있다는 과정을 보여주면 됨

*** 소스 코드**

```
//2021111945 송하연

#include <iostream>

using namespace std;
int level = 1;

void PrintArray(int* A, int n, int level) {

    cout << level <<"단계 : ";
    for (int i = 0; i < n; i++) {
        cout  <<"["<< A[i] <<"]" << " ";
    }
    cout << endl;
}


//비순환적 버블
void Bubble(int* A, int n){
    bool Sorted = false;
     while(!Sorted){
       Sorted = true;
       for(int i = 0; i < n-1; i++){
          if(A[i] > A[i+1]){
             swap(A[i], A[i+1]);
             Sorted = false;
          }
            PrintArray(A,n,level);
       }
       level++;
    }


}


//순환적 버블
void RecursiveBubble(int* A, int n){

   if (n <=  1 ) return;
   bool Sorted = true;

   for(int i = 0; i < n-1; i++){
       if(A[i] > A[i+1]){
          swap(A[i], A[i+1]);
          Sorted = false;
       }
```

```
            PrintArray(A,n,level);
        }
        level ++;
        if (Sorted) return;
        RecursiveBubble(A, n);
}


int main() {
    cout << "=========== 2021111945 송하연 ===========" << endl;
    int size;
    cout << "배열의 크기를 입력하세요 : " ;
    cin >>  size;
    int* A = new int[size];

    cout << "배열의 요소를 입력하세요(" << size << "개 입력) : ";
    for(int j = 0; j < size; j++){
        cin >> A[j];
    }

int choice;
    cout << "정렬 방법을 선택하세요 (1: 비순환적, 2: 순환적) : ";
    cin >> choice;

    if (choice == 1) {
        cout << "=========== 비순환적 버블 정렬 ===========" << endl;
        Bubble(A, size);
    } else if (choice == 2) {
        cout << "=========== 순환적 버블 정렬 ===========" << endl;
        RecursiveBubble(A, size);
    }
    delete[] A;

    return 0;
}
```

```
//2021111945 송하연

#include <iostream>
#include <stack>
#include<algorithm>
#include <set>

using namespace std;
const int MAX_LENTH = 10;

void GenerateRandomArray(int N[], int size) {
    srand(static_cast<unsigned int>(time(0)));
```

```
    set<int> uniqueNumbers;

    while (uniqueNumbers.size() < size) {
        int randomNum = rand() % 100000 + 1;
        uniqueNumbers.insert(randomNum);
    }

    int index = 0;
    for (int num : uniqueNumbers) {
        N[index++] = num;
    }
}

void PrintArray(int A[], int n) {
    for (int i = 0; i < n; i++) {
        cout << "[" << A[i] << "]" << " ";
    }
    cout << endl;
}


void PrintStep(int A[], int n, int pivotIndex, int left, int right) {
    cout << "배열 상태: ";
    PrintArray(A, n);
    cout << "선택된 피벗: [" << A[pivotIndex] << "] " <<  " Left: " << left << "
Right: " << right << endl << endl;
}


int Partition(int A[], int Left, int Right) {
    int PartElem = Left;
    int Value = A[PartElem];

    while (Left < Right) {
        while (Left < Right && A[Right] <= Value) { //pivot보다 큰 값이 나올 때까지
            Right--;
        }
        while (Left < Right && A[Left] >= Value) { //pivot보다 작은 값이 나올 때까지
            Left++;
        }
        if (Left < Right) {
            swap(A[Left], A[Right]);
        }
    }
    swap(A[PartElem], A[Right]);
    return Right;
}

void RecursiveQuickSort(int A[], int Left, int Right) {
```

```
    if (Left < Right) {
        int PartIndex = Partition(A, Left, Right);
        PrintStep(A, Right + 1, PartIndex, Left, Right);
        RecursiveQuickSort(A, Left, PartIndex - 1);
        RecursiveQuickSort(A, PartIndex + 1, Right);
    }
}

void QuickSort(int A[], int n){
    stack<int> s;
    s.push(0);
    s.push(n-1);
    while(!s.empty()){
        int Right = s.top(); s.pop();
        int Left = s.top(); s.pop();

        if (Left < Right) {
        int PartIndex = Partition(A,Left,Right);
        PrintStep(A, n, PartIndex, Left, Right);
         if (PartIndex - 1 > Left) {
               s.push(Left);
               s.push(PartIndex - 1);
            }
        if (PartIndex + 1 < Right) {
               s.push(PartIndex + 1);
               s.push(Right);
            }
        }
    }

}

int main() {
    cout << "\n=========== 2021111945 송하연 ===========" << endl << endl;
    int A[MAX_LENTH];

    cout << "\n=========== 정렬 전 배열  ===========" << endl;
    GenerateRandomArray(A, MAX_LENTH);
    PrintArray(A, MAX_LENTH);

    cout << "\n=========== 순환적 퀵 정렬 ===========" << endl;
    RecursiveQuickSort(A, 0, 9);

    cout << "\n=========== 순환적 퀵 정렬 최종 결과 ===========" << endl;
    PrintArray(A, MAX_LENTH);

    cout << "\n=========== 정렬 전 배열 ===========" << endl;
    GenerateRandomArray(A, MAX_LENTH);
    PrintArray(A, MAX_LENTH);
```

```
    cout << "\n=========== 비순환적 퀵 정렬 ===========" << endl;
    QuickSort(A, MAX_LENTH);

    cout << "\n=========== 비순환적 퀵 정렬 최종 결과 ===========" << endl;
    PrintArray(A, MAX_LENTH);

    return 0;
}
```