Các Thuật Toán Sắp Xếp

MSSV	20280083
Tên	Lại Toàn Thắng

Selection sort

- *Phân tích*

- Dùng vòng lặp để tìm phần tử nhỏ nhất trong dãy hiện hành.
- Đảo phần tử đó ra đầu mảng

1. Biên dịch đoạn chương trình nêu trên.
2. Tại sao trong hàm SelectionSort, vòng lặp thứ nhất có điều kiện là i < N-1?

Vì trong vòng lặp đó, min sẽ gán bằng i và đem min đi so sánh với các anh đằng sau min.

Đặt trường hợp min = i = n - 1 thì không còn anh nào ở phía sau để so sánh với min. Cụ thể là a[min] > a[n] (a[n] không tồn tại)

```
for (int i = 0; i < n-1; i++) // Vòng lặp này để move qua
// những ptu (đã được sắp xếp - Sorted Array)
{
```

3. Trả lời các dòng lệnh có yêu cầu ghi chú.
4. Sửa lại chương trình để nhập dãy số nguyên từ file input.txt có nội dung như sau:

5 1 2 3 8 6 23 10

```
ifstream fin;
fin.open("input.txt");
if (!fin.is_open())
{
	cout << "Cannot open this file\n";
	return -1;
}
int a[50];
int n = 0;
while (!fin.eof())
{
	fin >> a[n];
	n++;
}
```

5. Sửa hàm SelectionSort trên để sắp xếp dãy số nguyên ở câu 4 giảm dần.

```
25		if (a[j] > a[max])
```

Heap Sort

- *Phân tích*
- Hiệu chỉnh dãy số ban đầu về dạng heap được định nghĩa trên mảng (hay list).
- Áp dụng thuật toán Heap Sort trên cấu trúc này.

1. Trả lời các dòng lệnh có yêu cầu ghi chú.
2. Cho biết chức năng của đoạn chương trình trên.

3. Viết hàm void CreateHeap(int a[], int N); để chuyển đổi dãy $a_0, a_1, \ldots, a_{N-1}$ thành heap.

Gợi ý: Sử dụng hàm **Shift** bên trên với left hiện hành là phần tử ở giữa dãy (**(N-1)/2**). Lặp lại quá trình trên với left giảm dần về đầu dãy.

```
void CreateHeap(int a[], int n) // Build Heap
{
    int left;
    left = n / 2 - 1; //
    while (left >= 0)
    {
        Shift(a, left, n - 1); // Create Max Heap to sort elements
        left = left - 1;
    }
}
```

4. Viết hàm void HeapSort(int a[], int N); để sắp xếp một dãy số nguyên tăng dần.

Gợi ý: *Giai đoạn 1*: Hiệu chỉnh dãy ban đầu thành heap

Giai đoạn 2: Sắp xếp dãy số dựa trên heap.

- Xét dãy hiện hành là dãy nhập
- Hoán vị phần tử lớn nhất (a_0) về vị trí cuối.
- Xét dãy hiện hành loại đã trừ phần tử cuối.
- Hiệu chỉnh lại dãy hiện hành thành heap
- Lặp lại quá trình trên tới hết dãy ban đầu.

```
void HeapSort (int a[], int n)
{
    int right;
    CreateHeap(a, n);                       // Gđ1: Hiệu chỉnh dãy ban đầu thành heap
    right = n - 1;                          // right là vị trí phần tử cuối cùng
    while (right > 0)
    {
        swap(a[0], a[right]); // đổi vị trí phần tử đầu (là max) và phần tử index cuối
        right = right - 1; // Remove node cuối (Cố định nó)
        if (right > 0)
            Shift(a, 0, right); // Xử lí mảng hiện hành sau khi remove node lớn nhất, hiệu chỉnh
    }
}
```

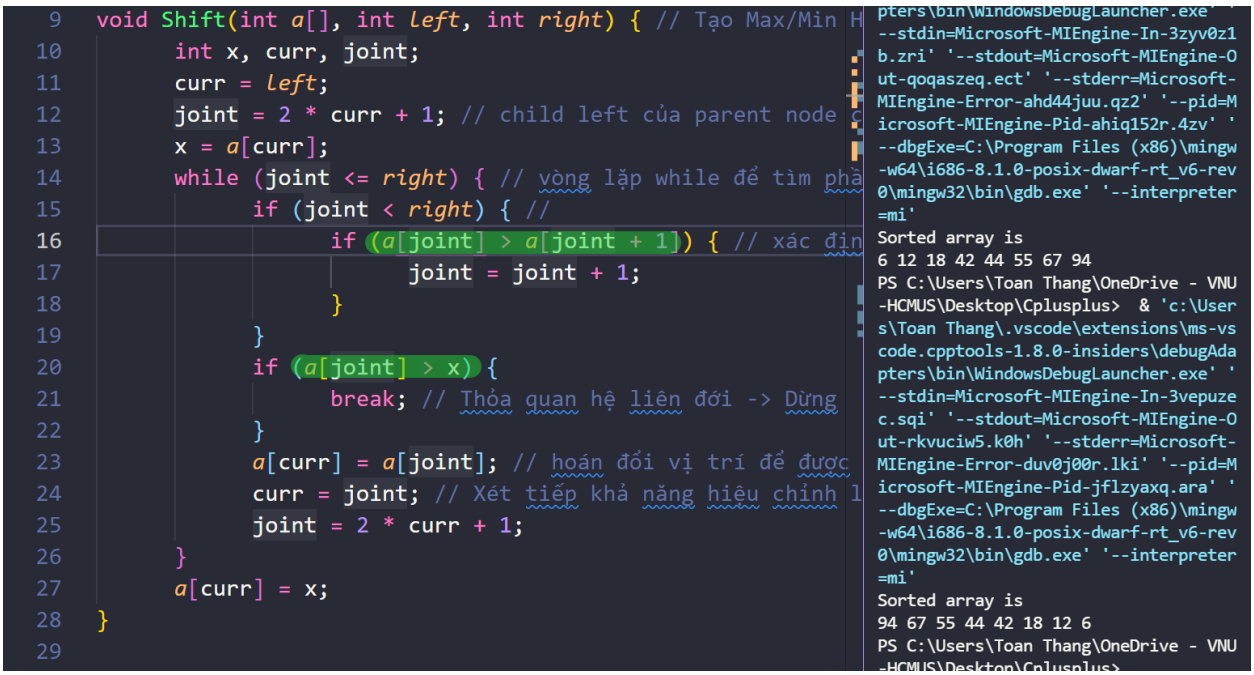
5. Bổ sung các hàm trên vào chương trình mẫu (CacThuatToanSapXep) đồng thời thay đổi hàm main và file input để sắp xếp dãy số nguyên sau tăng dần:

44 55 12 42 94 18 6 67

```
{
    ifstream fin;
    fin.open("input.txt");
    if (!fin.is_open())
    {
        cout << "cannot open this file\n";
        return -1;
    }
    int n = 0;
    int a[50];
    while (!fin.eof())
    {
        fin >> a[n];
        n++;
    }
    // int n = sizeof(a) / sizeof(a[0]);
    HeapSort(a, n);
    cout << "Sorted array is \n";
    printArray(a, n);
    return 0;
}
```

```
pters\bin\WindowsDebugLauncher.exe' '
--stdin=Microsoft-MIEngine-In-igh2pgw
f.rx2' '--stdout=Microsoft-MIEngine-O
ut-vf2oiymd.hic' '--stderr=Microsoft-
MIEngine-Error-1u5bhxah.1lo' '--pid=M
icrosoft-MIEngine-Pid-doscavz4.vxw' '
--dbgExe=C:\Program Files (x86)\mingw
-w64\i686-8.1.0-posix-dwarf-rt_v6-rev
0\mingw32\bin\gdb.exe' '--interpreter
=mi'
Sorted array is
44 55 12 42 94 18 6 67
PS C:\Users\Toan Thang\OneDrive - VNU
-HCMUS\Desktop\Cplusplus>  & 'c:\User
s\Toan Thang\.vscode\extensions\ms-vs
code.cpptools-1.8.0-insiders\debugAda
pters\bin\WindowsDebugLauncher.exe' '
--stdin=Microsoft-MIEngine-In-3zyv0z1
b.zri' '--stdout=Microsoft-MIEngine-O
ut-qoqaszeq.ect' '--stderr=Microsoft-
MIEngine-Error-ahd44juu.qz2' '--pid=M
icrosoft-MIEngine-Pid-ahiq152r.4zv' '
--dbgExe=C:\Program Files (x86)\mingw
-w64\i686-8.1.0-posix-dwarf-rt_v6-rev
0\mingw32\bin\gdb.exe' '--interpreter
=mi'
Sorted array is
6 12 18 42 44 55 67 94
PS C:\Users\Toan Thang\OneDrive - VNU
-HCMUS\Desktop\Cplusplus>
```

6. Viết lại thuật toán Heap Sort để sắp xếp dãy số ở câu 3 giảm dần.



```
9   void Shift(int a[], int left, int right) { // Tạo Max/Min H
10      int x, curr, joint;
11      curr = left;
12      joint = 2 * curr + 1; // child left của parent node
13      x = a[curr];
14      while (joint <= right) { // vòng lặp while để tìm phầ
15          if (joint < right) { //
16              if (a[joint] > a[joint + 1]) { // xác định
17                  joint = joint + 1;
18              }
19          }
20          if (a[joint] > x) {
21              break; // Thỏa quan hệ liên đới -> Dừng
22          }
23          a[curr] = a[joint]; // hoán đổi vị trí để được
24          curr = joint; // Xét tiếp khả năng hiệu chỉnh l
25          joint = 2 * curr + 1;
26      }
27      a[curr] = x;
28  }
29
```

```
pters\bin\WindowsDebugLauncher.exe'
--stdin=Microsoft-MIEngine-In-3zyv0z1
b.zri' '--stdout=Microsoft-MIEngine-O
ut-qoqaszeq.ect' '--stderr=Microsoft-
MIEngine-Error-ahd44juu.qz2' '--pid=M
icrosoft-MIEngine-Pid-ahiq152r.4zv' '
--dbgExe=C:\Program Files (x86)\mingw
-w64\i686-8.1.0-posix-dwarf-rt_v6-rev
0\mingw32\bin\gdb.exe' '--interpreter
=mi'
Sorted array is
6 12 18 42 44 55 67 94
PS C:\Users\Toan Thang\OneDrive - VNU
-HCMUS\Desktop\Cplusplus>  & 'c:\User
s\Toan Thang\.vscode\extensions\ms-vs
code.cpptools-1.8.0-insiders\debugAda
pters\bin\WindowsDebugLauncher.exe' '
--stdin=Microsoft-MIEngine-In-3vepuze
c.sqi' '--stdout=Microsoft-MIEngine-O
ut-rkvuciw5.k0h' '--stderr=Microsoft-
MIEngine-Error-duv0j00r.lki' '--pid=M
icrosoft-MIEngine-Pid-jflzyaxq.ara' '
--dbgExe=C:\Program Files (x86)\mingw
-w64\i686-8.1.0-posix-dwarf-rt_v6-rev
0\mingw32\bin\gdb.exe' '--interpreter
=mi'
Sorted array is
94 67 55 44 42 18 12 6
PS C:\Users\Toan Thang\OneDrive - VNU
-HCMUS\Desktop\Cplusplus>
```

Quick Sort

- *Phân tích*

- Chọn phần tử làm mốc.
- Tiến hành phân hoạch dãy ban đầu thành 3 phần $a_k<x$ (1), $a_k=x$ (2) và $a_k>x$ (3) theo thứ tự.
- Lặp lại thao tác trên trên 2 đoạn (1) và (3)

- *Chương trình mẫu*

- *Yêu cầu*

1. Bổ sung các hàm trên vào chương trình mẫu (CacThuatToanSapXep) đồng thời thay đổi hàm main và file input để sắp xếp dãy số nguyên sau tăng dần:

42 23 74 11 65 58 94 36 99 87

2. Sửa lại chương trình để đếm số phép gán và số phép so sánh sự dụng trong hàm QuickSort.

```
Sorted array is
11 23 36 42 58 65 74 87 94 99
So phep gan: 59
So phep so sanh: 54
```

Merge Sort

- *Phân tích*

- Phân phối đều luân phiên các dãy con độ dài k từ mảng a vào hai mảng b và c.
- Trộn mảng b và mảng c vào mảng a.
- Lặp lại quá trình trên với k tăng gấp đôi đến khi k lớn hơn hay bằng chiều dài của dãy.

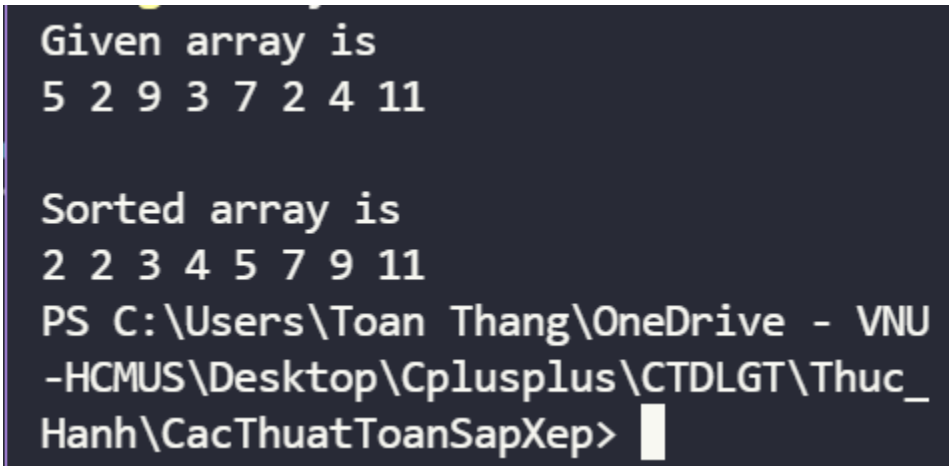
- *Chương trình mẫu*

- *Yêu cầu*

1. Trả lời các dòng lệnh có yêu cầu ghi chú.
2. Cho biết chức năng của từng hàm trên.
3. Bổ sung các hàm cần thiết vào chương trình mẫu (CacThuatToanSapXep) và viết hàm void MergeSort(int a[], int N); để sắp xếp dãy số nguyên sau tăng dần.

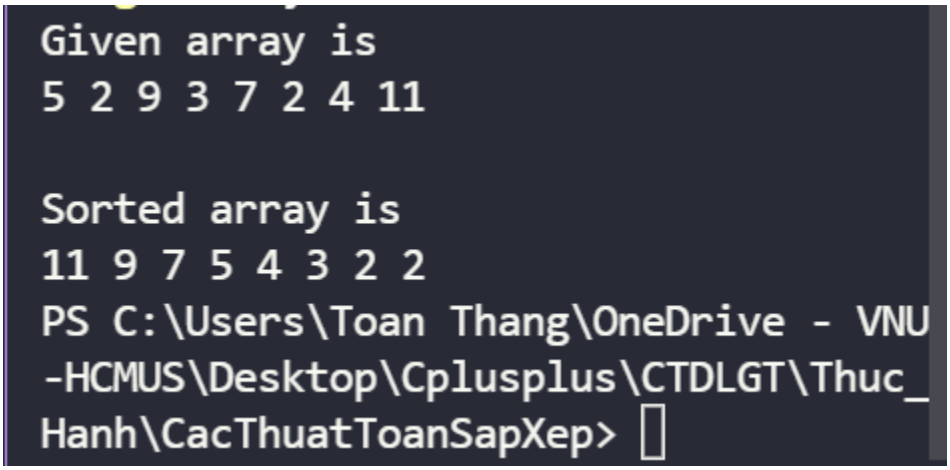
5 2 9 3 7 2 4 11

Gợi ý: Xem lại 2 thao tác đã nêu bên trên.



4. Sửa lại chương trình để sắp xếp dãy số trên giảm dần.

```
void MergeSubarr(int a[], int nb, int nc, int& pa, int& pb, int& pc, int k) {
    int rb, rc;
    rb = min(nb, pb + k);
    rc = min(nc, pb + k);
    while ((pb < rb) && (pc < rc)) {
        if (b[pb] > c[pc]) // Nếu sort giảm dần thì thay bằng b[pb] > c[pc]
            a[pa++] = b[pb++];
        else a[pa++] = c[pc++];
    }
    while (pb < rb) {
        a[pa++] = b[pb++];
    }
    while (pc < rc) {
        a[pa++] = c[pc++];
    }
}
```



5.Tìm hiểu và cài đặt thuật toán *Insertion Sort.*

6. Tìm hiểu và cài đặt thuật toán *Binary Insertion Sort.*

7. Tìm hiểu và cài đặt thuật toán *Interchange Sort.*

8. Tìm hiểu và cài đặt thuật toán *Bubble Sort.*

9. Tìm hiểu và cài đặt thuật toán *Shaker Sort.*

10. Tìm hiểu và cài đặt thuật toán *Shell Sort.*

11. Tìm hiểu và cài đặt thuật toán *Radix Sort* lần lượt trên 2 loại cấu trúc dữ liệu dạng mảng và dạng danh sách liên kết. So sánh và rút ra nhận xét.

Em làm và để trong folder nha thầy □□