ĐẠI HỌC QUỐC GIA

ĐẠI HỌC BÁCH KHOA TP HỒ CHÍ MINH

KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

BÀI TẬP LỚN MÔN KIẾN TRÚC MÁY TÍNH

Đề 3: Merge sort số thực chính xác đơn

LỚP L14 --- NHÓM 57 --- HK 231

Giảng viên hướng dẫn: Thầy Nguyễn Xuân Minh

Sinh viên thực hiện	Mã số sinh viên
Nguyễn Nhật Tân	2213059
Đào Duy Thành	2112288

Thành phố Hồ Chí Minh – 2023

Mục lục

I.Đề bài

Đề 3. Merge sort số thực chính xác đơn:

Viết chương trình sắp xếp dãy số thực chính xác đơn có 15 phần tử dùng giải thuật Merge sort. Yêu cầu xuất dãy ra màn hình mỗi bước. Dữ liệu đầu vào đọc từ file lưu trữ dạng nhị phân trên đĩa FLOAT15.BIN (15 phần tử x 4 bytes = 60 bytes).

II.Giải thuật merge sort

1. Khái niệm

Trong khoa học máy tính, sắp xếp trộn (merge sort) là một thuật toán sắp xếp để sắp xếp các danh sách (hoặc bất kỳ cấu trúc dữ liệu nào có thể truy cập tuần tự, v.d. luồng tập tin) theo một trật tự nào đó. Nó được xếp vào thể loại sắp xếp so sánh. Thuật toán này là một ví dụ tương đối điển hình của lối thuật toán chia để trị do John von Neumann đưa ra lần đầu năm 1945. Một thuật toán chi tiết được Goldstine và Neumann đưa ra năm 1948.

Giả sử có hai danh sách đã được sắp xếp a và b, ta có thể trộn chúng thành 1 danh sách mới c được sắp xếp như sau:

- So sánh hai phần tử đứng đầu của hai danh sách, lấy phần tử nhỏ hơn cho vào danh sách mới. Tiếp tục như vậy cho tới khi một trong hai danh sách là rỗng.
- Khi một trong hai danh sách là rỗng ta lấy phần còn lại của danh sách kia cho vào cuối danh sách mới

Ví dụ: Cho hai danh sách $\boldsymbol{a} = (\mathbf{1}, \mathbf{3}, \mathbf{7}, \mathbf{9}), \boldsymbol{b} = (\mathbf{2}, \mathbf{6})$ quá trình hòa nhập diễn ra như sau:

Danh sách a	Danh sách b	So sánh	Danh sách c
1,3,7,9	2,6	1<2	1
3,7,9	2,6	2<3	1,2
3,7,9	6	3<6	1,2,3
7,9	6	6<7	1,2,3,6
7,9			1,2,3,6,7,9

2. Một số cách thực hiện giải thuật merge sort:

Trộn từ dưới lên:

- Nếu danh sách con chỉ gồm hai phần tử, mỗi nửa của nó gồm một phần tử đương nhiên đã được sắp. Do đó việc trộn tại chố hai nửa danh sách này cho danh sách con 2 phân tử được sắp.
- Xuất phát từ đầu danh sách a ta trộn a[1] với a[2], a[3] với a[4],... Khi đó mọi danh sách con gồm hai phần tử của a đã được sắp. Tiếp tục trộn các danh sách con kế tiếp nhau gồm 2 phần tử thành các danh sách con 4 phần tử... Mỗi lần trộn số các danh sách con cần trộn giảm đi một nửa. Quá trình dừng lại khi số danh sách con chỉ còn một.

Ví dụ: Cho danh sách $\boldsymbol{a} = (\mathbf{2}, \mathbf{3}, \mathbf{5}, \mathbf{6}, \mathbf{4}, \mathbf{1}, \mathbf{7})$

Công việc	Số DS con	Kết quả
Trộn các phần tử đứng kề nhau	7	2,3-5,6-1,4-7
Trộn các danh sách con 2 phần tử kề nhau	4	2,3,5,6-1,4,7
Trộn các danh sách con 4 phần tử kề nhau	2	1,2,3,4,5,6,7

Trộn đệ quy:

- Một cách gọi đệ quy của sắp xếp trộn cũng thường được hướng dẫn trong các giáo trình giải thuật.
- Để sắp xếp trộn cho đoạn $\boldsymbol{a = [k_1..k_2]}$ ta chia đoạn đó thành 2 phần

$\boldsymbol{a = [k_1..k_3]}$ *và* $\boldsymbol{a = [k_3 + 1..k_2]}$ trong đó $\boldsymbol{k_3 = int((k_1 + k_2)/2)}$ tiến hành sắp xếp với mỗi phần rồi trộn chúng lại

Ví dụ: Cho danh sách $\boldsymbol{a = (37, 27, 43, 3, 9, 82, 10)}$

Giải thuật trộn đệ quy chia a thành hai danh sách con và tiến hành 3 bước

Danh sách trái	Danh sách phải
38,27,43,3	9,82,10

Sắp xếp trộn danh sách trái:

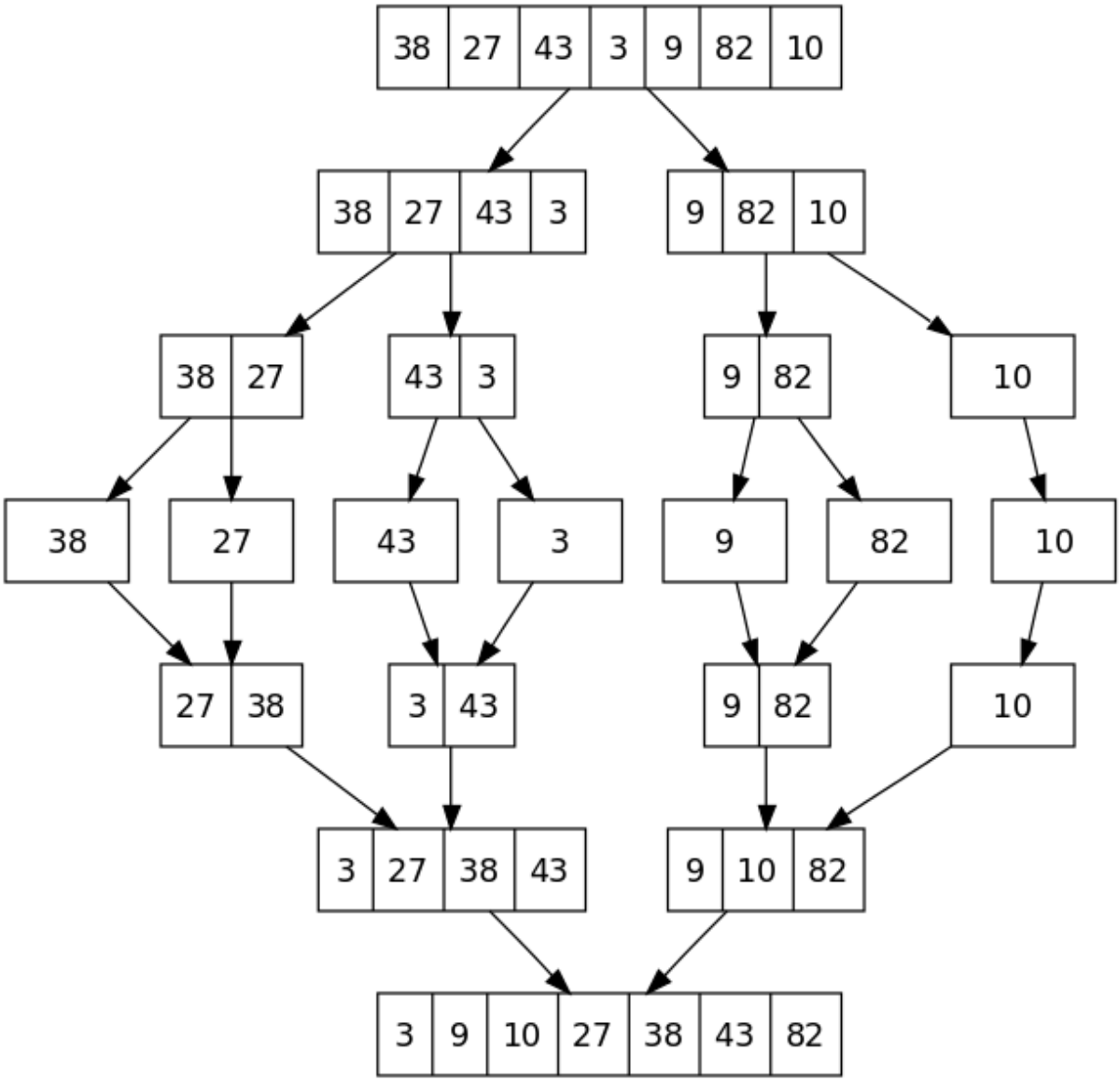
<table>
<tr><td colspan="4">Quá trình chia</td><td colspan="4">Quá trình trộn</td></tr>
<tr><td colspan="4">38,27,43,3</td><td>38</td><td>27</td><td>43</td><td>3</td></tr>
<tr><td colspan="2">38,27</td><td colspan="2">43,3</td><td colspan="2">27,38</td><td colspan="2">3,43</td></tr>
<tr><td>38</td><td>27</td><td>43</td><td>3</td><td colspan="4">3,27,38,43</td></tr>
</table>

Sắp xếp trộn danh sách phải:

<table>
<tr><td colspan="3">Quá trình chia</td><td colspan="3">Quá trình trộn</td></tr>
<tr><td colspan="3">9,82,10</td><td>9</td><td>82</td><td>10</td></tr>
<tr><td>9</td><td colspan="2">82,10</td><td>9</td><td colspan="2">10,82</td></tr>
<tr><td>9</td><td>82</td><td>10</td><td colspan="3">9,10,82</td></tr>
</table>

Trộn danh sách trái và danh sách phải:

Danh sách trái	Danh sách phải	Danh sách trộn
3,27,38,43	9,10,82	3,9,10,27,38,43,82



Hình minh họa cho ví dụ trên

III. Giải pháp hiện thực

3.1 Phác họa các bước trong quá trình code.

Trong quá trình này chúng ta sẽ chọn giải thuật sắp xếp **trộn đệ quy**, chia nhỏ thành 2 bước:

Bước 1: Tách dãy đã cho

- Chia dãy cần sắp xếp thành 2 dãy con.

- Từ các dãy con thu được tiếp tục chia thành 2 dãy con nhỏ hơn nữa.
- Quá trình trên được lặp lại cho đến khi dãy con cuối cùng chỉ còn 1 phần tử.

Bước 2: Gộp dãy đã cho

-Bắt đầu gộp 2 dãy con nhỏ nhất (1phần tử) sao cho các phần tử sẽ đúng thứ tự sắp xếp.

-Từ 2 dãy con vừa được gộp lại tiếp tục gộp thành dãy con lớn hơn sao cho cũng đã đúng thứ tự sắp xếp.

-Quá tình gộp này được lặp lại tiếp tục cho đến khi thu được một dãy đủ các phần tử như dãy ban đầu nhưng đã được sắp xếp.

3.2 Mã giả của giải thuật Merge Sort:

Ta có 2 hàm chính để thực hiện giải thuật này:

Hàm merge_sort: tách dãy cần sắp xếp thành 2 dãy con, rồi tiến hành gộp lại bằng hàm merge_array.

Hàm merge_array: Gộp 2 dãy đã được sắp xếp từ bé đến lớn thành 1 dãy được sắp xếp từ bé đến lớn.

3.2.1 Hàm merge_sort

1. lef t = 0, right = size - 1
2. if lef t < right then
    - middle = (lef t + right) / 2
    - merge_sort(lef t, middle)
    - merge_sort(middle + 1, right)
    - merge_array(lef t, middle, right)

3.2.2 Hàm merge_array

1. arr_temp1 <- [lef t, middle]

2. arr_temp2 <- [middle + 1, right]

3. N1 = middle − lef t + 1

4. N2 = right − middle

5. i = 0, j = 0, k = 0

6. while i < N1 AND j < N2 do

- if (arr_temp1[i] < arr_temp2[j] then

  (a) array[k] = arr_temp1[i]

  (b) ++k, ++i

  else

  (a) array[k] = arr_temp2[j]

  (b) ++k, ++j

7. while i < N1 do

- array[k] = arr_temp1[i]
- ++k, ++i

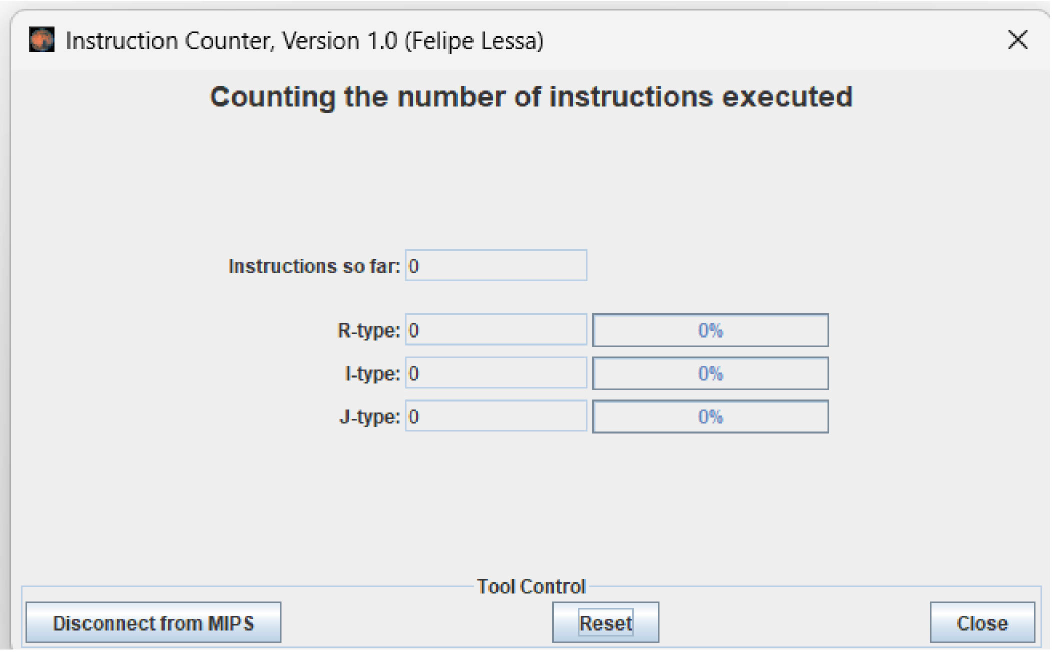
8. while j < N2 do

- array[k] = arr_temp2[j]
- ++k, ++j

9. print_step_array(lef t, right)

IV.Thống kê số lệnh, loại lệnh (instruction type) sử dụng trong chương trình.

Sử dụng công cụ Instruction Counter để đếm số lệnh trong chương trình:

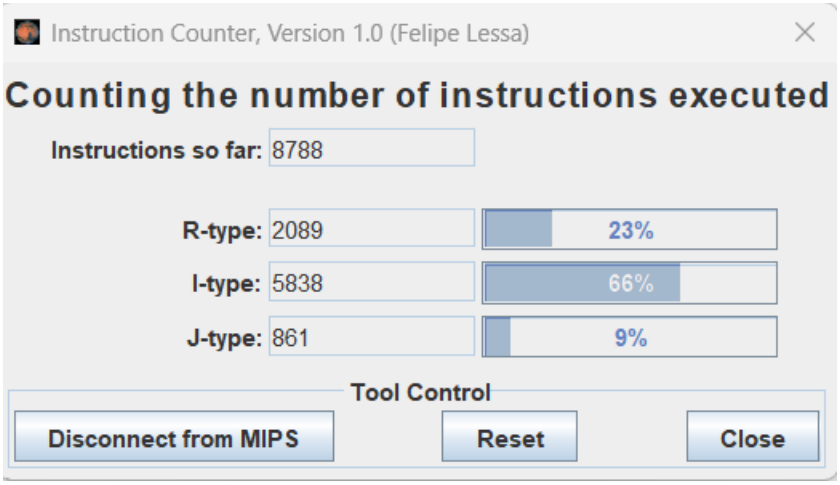


Hình 1: Hộp thoại công cụ Instruction Counter

Ta chạy chương trình merge_sort với dữ liệu đầu vào đọc từ file lưu trữ dạng nhị phân trên đĩa FLOAT15.BIN, sau khi thực thi ta được kết quả :

```
Before sorting:  -0.43125433 -5.43211 -4.4321213 -1.2309877 -4.4301233 -1.2245612 -4.525679 -13.432199 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -0.43125433 -5.43211 -4.4321213 -1.2309877 -4.4301233 -1.2245612 -4.525679 -13.432199 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -0.43125433 -5.43211 -4.4321213 -1.2309877 -4.4301233 -1.2245612 -4.525679 -13.432199 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -5.43211 -0.43125433 -4.4321213 -1.2309877 -4.4301233 -1.2245612 -4.525679 -13.432199 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -5.43211 -0.43125433 -4.4321213 -1.2309877 -4.4301233 -1.2245612 -4.525679 -13.432199 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -5.43211 -0.43125433 -4.4321213 -1.2309877 -4.4301233 -1.2245612 -4.525679 -13.432199 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -5.43211 -0.43125433 -4.4321213 -1.2309877 -4.4301233 -1.2245612 -4.525679 -13.432199 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -5.43211 -0.43125433 -4.4321213 -1.2309877 -4.4301233 -1.2245612 -4.525679 -13.432199 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -5.43211 -0.43125433 -4.4321213 -1.2309877 -4.4301233 -1.2245612 -4.525679 -13.432199 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -5.43211 -4.4321213 -1.2309877 -0.43125433 -4.4301233 -1.2245612 -4.525679 -13.432199 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -5.43211 -4.4321213 -1.2309877 -0.43125433 -4.4301233 -1.2245612 -4.525679 -13.432199 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -5.43211 -4.4321213 -1.2309877 -0.43125433 -4.4301233 -1.2245612 -4.525679 -13.432199 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -5.43211 -4.4321213 -1.2309877 -0.43125433 -4.4301233 -1.2245612 -4.525679 -13.432199 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -5.43211 -4.4321213 -1.2309877 -0.43125433 -4.4301233 -1.2245612 -4.525679 -13.432199 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -5.43211 -4.4321213 -1.2309877 -0.43125433 -4.4301233 -1.2245612 -4.525679 -13.432199 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -5.43211 -4.4321213 -1.2309877 -0.43125433 -4.4301233 -1.2245612 -4.525679 -13.432199 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -5.43211 -4.4321213 -1.2309877 -0.43125433 -4.4301233 -1.2245612 -4.525679 -13.432199 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -5.43211 -4.4321213 -1.2309877 -0.43125433 -4.4301233 -1.2245612 -13.432199 -4.525679 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -5.43211 -4.4321213 -1.2309877 -0.43125433 -4.4301233 -1.2245612 -13.432199 -4.525679 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -5.43211 -4.4321213 -1.2309877 -0.43125433 -13.432199 -4.525679 -4.4301233 -1.2245612 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -5.43211 -4.4321213 -1.2309877 -0.43125433 -13.432199 -4.525679 -4.4301233 -1.2245612 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -13.432199 -5.43211 -4.525679 -4.4321213 -4.4301233 -1.2309877 -1.2245612 -0.43125433 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -13.432199 -5.43211 -4.525679 -4.4321213 -4.4301233 -1.2309877 -1.2245612 -0.43125433 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -13.432199 -5.43211 -4.525679 -4.4321213 -4.4301233 -1.2309877 -1.2245612 -0.43125433 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -13.432199 -5.43211 -4.525679 -4.4321213 -4.4301233 -1.2309877 -1.2245612 -0.43125433 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -13.432199 -5.43211 -4.525679 -4.4321213 -4.4301233 -1.2309877 -1.2245612 -0.43125433 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -13.432199 -5.43211 -4.525679 -4.4321213 -4.4301233 -1.2309877 -1.2245612 -0.43125433 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -13.432199 -5.43211 -4.525679 -4.4321213 -4.4301233 -1.2309877 -1.2245612 -0.43125433 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -13.432199 -5.43211 -4.525679 -4.4321213 -4.4301233 -1.2309877 -1.2245612 -0.43125433 -1.4213457 -0.22132123 -3.9204123 -4.4321213 -6.4343214 -5.1243567 -12.1456785
 -13.432199 -5.43211 -4.525679 -4.4321213 -4.4301233 -1.2309877 -1.2245612 -0.43125433 -1.4213457 -0.22132123 -4.4321213 -3.9204123 -6.4343214 -5.1243567 -12.1456785
 -13.432199 -5.43211 -4.525679 -4.4321213 -4.4301233 -1.2309877 -1.2245612 -0.43125433 -1.4213457 -0.22132123 -4.4321213 -3.9204123 -6.4343214 -5.1243567 -12.1456785
 -13.432199 -5.43211 -4.525679 -4.4321213 -4.4301233 -1.2309877 -1.2245612 -0.43125433 -4.4321213 -3.9204123 -1.4213457 -0.22132123 -6.4343214 -5.1243567 -12.1456785
 -13.432199 -5.43211 -4.525679 -4.4321213 -4.4301233 -1.2309877 -1.2245612 -0.43125433 -4.4321213 -3.9204123 -1.4213457 -0.22132123 -6.4343214 -5.1243567 -12.1456785
 -13.432199 -5.43211 -4.525679 -4.4321213 -4.4301233 -1.2309877 -1.2245612 -0.43125433 -4.4321213 -3.9204123 -1.4213457 -0.22132123 -6.4343214 -5.1243567 -12.1456785
 -13.432199 -5.43211 -4.525679 -4.4321213 -4.4301233 -1.2309877 -1.2245612 -0.43125433 -4.4321213 -3.9204123 -1.4213457 -0.22132123 -6.4343214 -5.1243567 -12.1456785
 -13.432199 -5.43211 -4.525679 -4.4321213 -4.4301233 -1.2309877 -1.2245612 -0.43125433 -4.4321213 -3.9204123 -1.4213457 -0.22132123 -6.4343214 -5.1243567 -12.1456785
 -13.432199 -5.43211 -4.525679 -4.4321213 -4.4301233 -1.2309877 -1.2245612 -0.43125433 -4.4321213 -3.9204123 -1.4213457 -0.22132123 -6.4343214 -5.1243567 -12.1456785
 -13.432199 -5.43211 -4.525679 -4.4321213 -4.4301233 -1.2309877 -1.2245612 -0.43125433 -4.4321213 -3.9204123 -1.4213457 -0.22132123 -6.4343214 -5.1243567 -12.1456785
 -13.432199 -5.43211 -4.525679 -4.4321213 -4.4301233 -1.2309877 -1.2245612 -0.43125433 -4.4321213 -3.9204123 -1.4213457 -0.22132123 -12.1456785 -6.4343214 -5.1243567


 -13.432199 -5.43211 -4.525679 -4.4321213 -4.4301233 -1.2309877 -1.2245612 -0.43125433 -4.4321213 -3.9204123 -1.4213457 -0.22132123 -12.1456785 -6.4343214 -5.1243567
 -13.432199 -5.43211 -4.525679 -4.4321213 -4.4301233 -1.2309877 -1.2245612 -0.43125433 -12.1456785 -6.4343214 -5.1243567 -4.4321213 -3.9204123 -1.4213457 -0.22132123
 -13.432199 -5.43211 -4.525679 -4.4321213 -4.4301233 -1.2309877 -1.2245612 -0.43125433 -12.1456785 -6.4343214 -5.1243567 -4.4321213 -3.9204123 -1.4213457 -0.22132123
 -13.432199 -12.1456785 -6.4343214 -5.43211 -5.1243567 -4.525679 -4.4321213 -4.4321213 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -1.2245612 -0.43125433 -0.22132123
 After sorting:  -13.432199 -12.1456785 -6.4343214 -5.43211 -5.1243567 -4.525679 -4.4321213 -4.4321213 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -1.2245612 -0.43125433 -0.22132123
```



Hình 2: Kết quả chạy thử công cự Instruction Counter

Có 8788 lệnh trong chương trình này, trong đó có:

- R-type: 2089 chiếm 23%
- I-type: 5838 chiếm 66%
- j-type: 861 chiếm 9%

V.Thời gian chạy của chương trình (CR=1GHz).

Sử dụng công thức sau để tính thời gian chạy:

$$\textbf{CPU time} = \frac{\boldsymbol{CPU\ Clock\ cycle}}{\boldsymbol{Clock\ rate}} = \frac{\boldsymbol{Instruction\ Count} \times \boldsymbol{CPI}}{\boldsymbol{Clock\ rate}}$$

Trong đó:

- CPU time là thời gian xử lý của chương trình (Không tính thời gian giao tiếp I/O, thời gian chờ...).
- CPU Clock cycles: Tổng số chu kỳ thực thi.
- Instuction Count là tổng số lệnh thực thi của chương trình.
- CPI(Cycle per Instruction) là số chu kỳ thực hiện trên một lệnh.
- Clock rate là số chu kỳ thực thi trên một giây hay còn gọi là tần số, VD: 1 GHz: có nghĩa là trong một giây có $\mathbf{1} \times \mathbf{10^9}$ dao động

Xét Clock rate = 1GHz và CPI = 1, với ví dụ ở trên ta có

Instruction Count = 5847

$$\textbf{CPU time} = \frac{\boldsymbol{Instruction\ Count\ \times CPI}}{\boldsymbol{Clock\ rate}} = \frac{\mathbf{8788 \times 1}}{\mathbf{1 \times\ 10^9}} = \boldsymbol{8,788us}$$

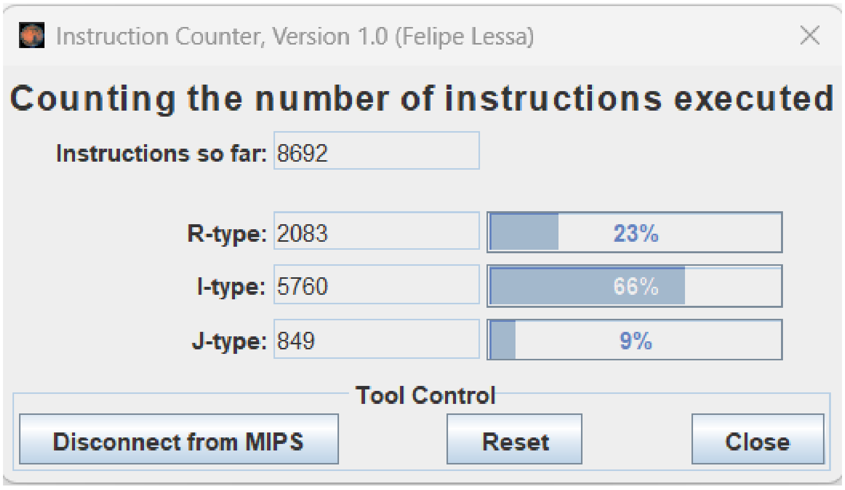
VI.Kết quả kiểm thử:

Một số kết quả của bài toán khi chạy với các input khác nhau:

Testcase 1: 7.890123, 2.345678, 11.234567, 4.567890, 0.123456, 13.456789, 14.567890, 1.234567, 5.678901, 12.345678, 3.456789, 10.123456, 8.901234, 9.012345, 6.789012

```
Before sorting:  7.890123 2.345678 11.234567 4.56789 0.123456 13.456789 14.56789 1.234567 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 7.890123 2.345678 11.234567 4.56789 0.123456 13.456789 14.56789 1.234567 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 7.890123 2.345678 11.234567 4.56789 0.123456 13.456789 14.56789 1.234567 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 2.345678 7.890123 11.234567 4.56789 0.123456 13.456789 14.56789 1.234567 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 2.345678 7.890123 11.234567 4.56789 0.123456 13.456789 14.56789 1.234567 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 2.345678 7.890123 11.234567 4.56789 0.123456 13.456789 14.56789 1.234567 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 2.345678 7.890123 11.234567 4.56789 0.123456 13.456789 14.56789 1.234567 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 2.345678 7.890123 4.56789 11.234567 0.123456 13.456789 14.56789 1.234567 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 2.345678 7.890123 4.56789 11.234567 0.123456 13.456789 14.56789 1.234567 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 2.345678 4.56789 7.890123 11.234567 0.123456 13.456789 14.56789 1.234567 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 2.345678 4.56789 7.890123 11.234567 0.123456 13.456789 14.56789 1.234567 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 2.345678 4.56789 7.890123 11.234567 0.123456 13.456789 14.56789 1.234567 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 2.345678 4.56789 7.890123 11.234567 0.123456 13.456789 14.56789 1.234567 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 2.345678 4.56789 7.890123 11.234567 0.123456 13.456789 14.56789 1.234567 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 2.345678 4.56789 7.890123 11.234567 0.123456 13.456789 14.56789 1.234567 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 2.345678 4.56789 7.890123 11.234567 0.123456 13.456789 14.56789 1.234567 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 2.345678 4.56789 7.890123 11.234567 0.123456 13.456789 14.56789 1.234567 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 2.345678 4.56789 7.890123 11.234567 0.123456 13.456789 1.234567 14.56789 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 2.345678 4.56789 7.890123 11.234567 0.123456 13.456789 1.234567 14.56789 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 2.345678 4.56789 7.890123 11.234567 0.123456 1.234567 13.456789 14.56789 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 2.345678 4.56789 7.890123 11.234567 0.123456 1.234567 13.456789 14.56789 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 0.123456 1.234567 2.345678 4.56789 7.890123 11.234567 13.456789 14.56789 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 0.123456 1.234567 2.345678 4.56789 7.890123 11.234567 13.456789 14.56789 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 0.123456 1.234567 2.345678 4.56789 7.890123 11.234567 13.456789 14.56789 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 0.123456 1.234567 2.345678 4.56789 7.890123 11.234567 13.456789 14.56789 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 0.123456 1.234567 2.345678 4.56789 7.890123 11.234567 13.456789 14.56789 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 0.123456 1.234567 2.345678 4.56789 7.890123 11.234567 13.456789 14.56789 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 0.123456 1.234567 2.345678 4.56789 7.890123 11.234567 13.456789 14.56789 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 0.123456 1.234567 2.345678 4.56789 7.890123 11.234567 13.456789 14.56789 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 0.123456 1.234567 2.345678 4.56789 7.890123 11.234567 13.456789 14.56789 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 0.123456 1.234567 2.345678 4.56789 7.890123 11.234567 13.456789 14.56789 5.678901 12.345678 3.456789 10.123456 8.901234 9.012345 6.789012
 0.123456 1.234567 2.345678 4.56789 7.890123 11.234567 13.456789 14.56789 3.456789 5.678901 10.123456 12.345678 8.901234 9.012345 6.789012
 0.123456 1.234567 2.345678 4.56789 7.890123 11.234567 13.456789 14.56789 3.456789 5.678901 10.123456 12.345678 8.901234 9.012345 6.789012
 0.123456 1.234567 2.345678 4.56789 7.890123 11.234567 13.456789 14.56789 3.456789 5.678901 10.123456 12.345678 8.901234 9.012345 6.789012
 0.123456 1.234567 2.345678 4.56789 7.890123 11.234567 13.456789 14.56789 3.456789 5.678901 10.123456 12.345678 8.901234 9.012345 6.789012
 0.123456 1.234567 2.345678 4.56789 7.890123 11.234567 13.456789 14.56789 3.456789 5.678901 10.123456 12.345678 8.901234 9.012345 6.789012
 0.123456 1.234567 2.345678 4.56789 7.890123 11.234567 13.456789 14.56789 3.456789 5.678901 10.123456 12.345678 8.901234 9.012345 6.789012
 0.123456 1.234567 2.345678 4.56789 7.890123 11.234567 13.456789 14.56789 3.456789 5.678901 10.123456 12.345678 8.901234 9.012345 6.789012
 0.123456 1.234567 2.345678 4.56789 7.890123 11.234567 13.456789 14.56789 3.456789 5.678901 10.123456 12.345678 6.789012 8.901234 9.012345
```

```
 0.123456 1.234567 2.345678 4.56789 7.890123 11.234567 13.456789 14.56789 3.456789 5.678901 10.123456 12.345678 6.789012 8.901234 9.012345
 0.123456 1.234567 2.345678 4.56789 7.890123 11.234567 13.456789 14.56789 3.456789 5.678901 6.789012 8.901234 9.012345 10.123456 12.345678
 0.123456 1.234567 2.345678 4.56789 7.890123 11.234567 13.456789 14.56789 3.456789 5.678901 6.789012 8.901234 9.012345 10.123456 12.345678
 0.123456 1.234567 2.345678 3.456789 4.56789 5.678901 6.789012 7.890123 8.901234 9.012345 10.123456 11.234567 12.345678 13.456789 14.56789
 After sorting:  0.123456 1.234567 2.345678 3.456789 4.56789 5.678901 6.789012 7.890123 8.901234 9.012345 10.123456 11.234567 12.345678 13.456789 14.56789
-- program is finished running --
```

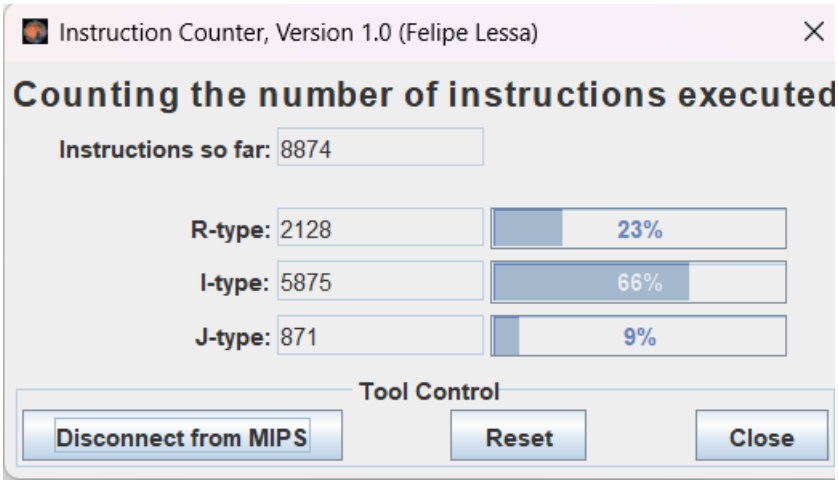


$$\textbf{CPU time} = \frac{\boldsymbol{Instruction\ Count\ \times CPI}}{\boldsymbol{Clock\ rate}} = \frac{\mathbf{8692 \times 1}}{\mathbf{1 \times\ 10^{9}}} = \mathbf{8.692}\boldsymbol{us}$$

Testcase 2: 6.43432, 1.42134, 4.52567, 0.22132, 4.53201, -1.22456, 5.43210, 4.43012, 1.23098, 5.12435, -13.43221, -4.43212, 0.43125, -12.14567, 3.92041

```
Before sorting:  6.43432 1.42134 4.52567 0.22132 4.53201 -1.22456 5.4321 4.43012 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 6.43432 1.42134 4.52567 0.22132 4.53201 -1.22456 5.4321 4.43012 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 6.43432 1.42134 4.52567 0.22132 4.53201 -1.22456 5.4321 4.43012 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 1.42134 6.43432 4.52567 0.22132 4.53201 -1.22456 5.4321 4.43012 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 1.42134 6.43432 4.52567 0.22132 4.53201 -1.22456 5.4321 4.43012 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 1.42134 6.43432 4.52567 0.22132 4.53201 -1.22456 5.4321 4.43012 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 1.42134 6.43432 4.52567 0.22132 4.53201 -1.22456 5.4321 4.43012 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 1.42134 6.43432 0.22132 4.52567 4.53201 -1.22456 5.4321 4.43012 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 1.42134 6.43432 0.22132 4.52567 4.53201 -1.22456 5.4321 4.43012 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 0.22132 1.42134 4.52567 6.43432 4.53201 -1.22456 5.4321 4.43012 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 0.22132 1.42134 4.52567 6.43432 4.53201 -1.22456 5.4321 4.43012 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 0.22132 1.42134 4.52567 6.43432 4.53201 -1.22456 5.4321 4.43012 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 0.22132 1.42134 4.52567 6.43432 4.53201 -1.22456 5.4321 4.43012 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 0.22132 1.42134 4.52567 6.43432 -1.22456 4.53201 5.4321 4.43012 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 0.22132 1.42134 4.52567 6.43432 -1.22456 4.53201 5.4321 4.43012 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 0.22132 1.42134 4.52567 6.43432 -1.22456 4.53201 5.4321 4.43012 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 0.22132 1.42134 4.52567 6.43432 -1.22456 4.53201 5.4321 4.43012 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 0.22132 1.42134 4.52567 6.43432 -1.22456 4.53201 4.43012 5.4321 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 0.22132 1.42134 4.52567 6.43432 -1.22456 4.53201 4.43012 5.4321 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 0.22132 1.42134 4.52567 6.43432 -1.22456 4.43012 4.53201 5.4321 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 0.22132 1.42134 4.52567 6.43432 -1.22456 4.43012 4.53201 5.4321 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 -1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 -1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 -1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 -1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 -1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 -1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 -1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 -1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 -1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 -1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 1.23098 5.12435 -13.43221 -4.43212 0.43125 -12.14567 3.92041
 -1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 -13.43221 -4.43212 1.23098 5.12435 0.43125 -12.14567 3.92041
 -1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 -13.43221 -4.43212 1.23098 5.12435 0.43125 -12.14567 3.92041
 -1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 -13.43221 -4.43212 1.23098 5.12435 0.43125 -12.14567 3.92041
 -1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 -13.43221 -4.43212 1.23098 5.12435 0.43125 -12.14567 3.92041
 -1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 -13.43221 -4.43212 1.23098 5.12435 -12.14567 0.43125 3.92041
 -1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 -13.43221 -4.43212 1.23098 5.12435 -12.14567 0.43125 3.92041
 -1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 -13.43221 -4.43212 1.23098 5.12435 -12.14567 0.43125 3.92041
 -1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 -13.43221 -4.43212 1.23098 5.12435 -12.14567 0.43125 3.92041
```

```
-1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 -13.43221 -4.43212 1.23098 5.12435 -12.14567 0.43125 3.92041
-1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 -13.43221 -4.43212 1.23098 5.12435 -12.14567 0.43125 3.92041
-1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 -13.43221 -12.14567 -4.43212 0.43125 1.23098 3.92041 5.12435
-1.22456 0.22132 1.42134 4.43012 4.52567 4.53201 5.4321 6.43432 -13.43221 -12.14567 -4.43212 0.43125 1.23098 3.92041 5.12435
-13.43221 -12.14567 -4.43212 -1.22456 0.22132 0.43125 1.23098 1.42134 3.92041 4.43012 4.52567 4.53201 5.12435 5.4321 6.43432
After sorting:  -13.43221 -12.14567 -4.43212 -1.22456 0.22132 0.43125 1.23098 1.42134 3.92041 4.43012 4.52567 4.53201 5.12435 5.4321 6.43432
-- program is finished running --
```

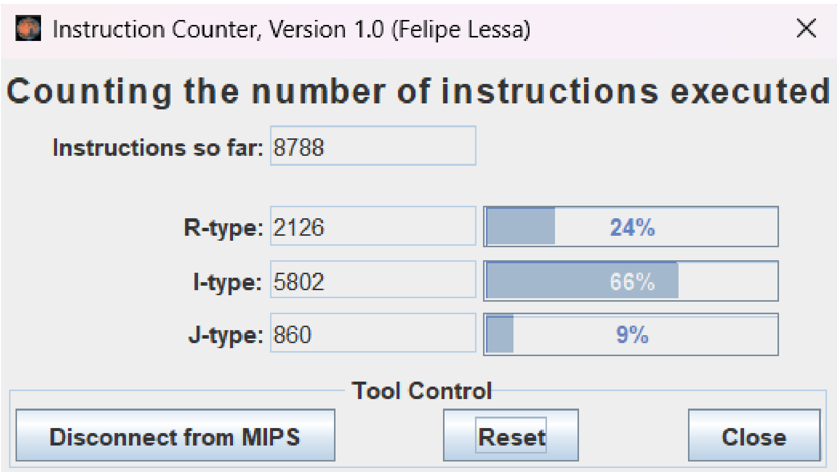


$$\textbf{CPU time} = \frac{\boldsymbol{Instruction\ Count\ \times CPI}}{\boldsymbol{Clock\ rate}} = \frac{\mathbf{8874 \times 1}}{\mathbf{1 \times\ 10^9}} = \mathbf{8.874}\boldsymbol{us}$$

Testcase 3: 0.43125, 5.43210, -4.43212, 1.23098, 4.43012, -1.22456, 4.52567, -13.43221, 1.42134, 0.22132, 3.92041, -4.43212, 6.43432, 5.12435, -12.14567

```
Before sorting:  0.43125 5.4321 -4.43212 1.23098 4.43012 -1.22456 4.52567 -13.43221 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 0.43125 5.4321 -4.43212 1.23098 4.43012 -1.22456 4.52567 -13.43221 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 0.43125 5.4321 -4.43212 1.23098 4.43012 -1.22456 4.52567 -13.43221 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 0.43125 5.4321 -4.43212 1.23098 4.43012 -1.22456 4.52567 -13.43221 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 0.43125 5.4321 -4.43212 1.23098 4.43012 -1.22456 4.52567 -13.43221 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 0.43125 5.4321 -4.43212 1.23098 4.43012 -1.22456 4.52567 -13.43221 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 0.43125 5.4321 -4.43212 1.23098 4.43012 -1.22456 4.52567 -13.43221 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 0.43125 5.4321 -4.43212 1.23098 4.43012 -1.22456 4.52567 -13.43221 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 0.43125 5.4321 -4.43212 1.23098 4.43012 -1.22456 4.52567 -13.43221 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 -4.43212 0.43125 1.23098 5.4321 4.43012 -1.22456 4.52567 -13.43221 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 -4.43212 0.43125 1.23098 5.4321 4.43012 -1.22456 4.52567 -13.43221 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 -4.43212 0.43125 1.23098 5.4321 4.43012 -1.22456 4.52567 -13.43221 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 -4.43212 0.43125 1.23098 5.4321 4.43012 -1.22456 4.52567 -13.43221 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 -4.43212 0.43125 1.23098 5.4321 -1.22456 4.43012 4.52567 -13.43221 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 -4.43212 0.43125 1.23098 5.4321 -1.22456 4.43012 4.52567 -13.43221 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 -4.43212 0.43125 1.23098 5.4321 -1.22456 4.43012 4.52567 -13.43221 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 -4.43212 0.43125 1.23098 5.4321 -1.22456 4.43012 4.52567 -13.43221 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 -4.43212 0.43125 1.23098 5.4321 -1.22456 4.43012 -13.43221 4.52567 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 -4.43212 0.43125 1.23098 5.4321 -1.22456 4.43012 -13.43221 4.52567 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 -4.43212 0.43125 1.23098 5.4321 -13.43221 -1.22456 4.43012 4.52567 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 -4.43212 0.43125 1.23098 5.4321 -13.43221 -1.22456 4.43012 4.52567 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 -13.43221 -4.43212 -1.22456 0.43125 1.23098 4.43012 4.52567 5.4321 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 -13.43221 -4.43212 -1.22456 0.43125 1.23098 4.43012 4.52567 5.4321 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 -13.43221 -4.43212 -1.22456 0.43125 1.23098 4.43012 4.52567 5.4321 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 -13.43221 -4.43212 -1.22456 0.43125 1.23098 4.43012 4.52567 5.4321 1.42134 0.22132 3.92041 -4.43212 6.43432 5.12435 -12.14567
 -13.43221 -4.43212 -1.22456 0.43125 1.23098 4.43012 4.52567 5.4321 0.22132 1.42134 3.92041 -4.43212 6.43432 5.12435 -12.14567
 -13.43221 -4.43212 -1.22456 0.43125 1.23098 4.43012 4.52567 5.4321 0.22132 1.42134 3.92041 -4.43212 6.43432 5.12435 -12.14567
 -13.43221 -4.43212 -1.22456 0.43125 1.23098 4.43012 4.52567 5.4321 0.22132 1.42134 3.92041 -4.43212 6.43432 5.12435 -12.14567
 -13.43221 -4.43212 -1.22456 0.43125 1.23098 4.43012 4.52567 5.4321 0.22132 1.42134 3.92041 -4.43212 6.43432 5.12435 -12.14567
 -13.43221 -4.43212 -1.22456 0.43125 1.23098 4.43012 4.52567 5.4321 0.22132 1.42134 -4.43212 3.92041 6.43432 5.12435 -12.14567
 -13.43221 -4.43212 -1.22456 0.43125 1.23098 4.43012 4.52567 5.4321 0.22132 1.42134 -4.43212 3.92041 6.43432 5.12435 -12.14567
 -13.43221 -4.43212 -1.22456 0.43125 1.23098 4.43012 4.52567 5.4321 -4.43212 0.22132 1.42134 3.92041 6.43432 5.12435 -12.14567
 -13.43221 -4.43212 -1.22456 0.43125 1.23098 4.43012 4.52567 5.4321 -4.43212 0.22132 1.42134 3.92041 6.43432 5.12435 -12.14567
 -13.43221 -4.43212 -1.22456 0.43125 1.23098 4.43012 4.52567 5.4321 -4.43212 0.22132 1.42134 3.92041 6.43432 5.12435 -12.14567
 -13.43221 -4.43212 -1.22456 0.43125 1.23098 4.43012 4.52567 5.4321 -4.43212 0.22132 1.42134 3.92041 6.43432 5.12435 -12.14567
 -13.43221 -4.43212 -1.22456 0.43125 1.23098 4.43012 4.52567 5.4321 -4.43212 0.22132 1.42134 3.92041 5.12435 6.43432 -12.14567
 -13.43221 -4.43212 -1.22456 0.43125 1.23098 4.43012 4.52567 5.4321 -4.43212 0.22132 1.42134 3.92041 5.12435 6.43432 -12.14567
 -13.43221 -4.43212 -1.22456 0.43125 1.23098 4.43012 4.52567 5.4321 -4.43212 0.22132 1.42134 3.92041 5.12435 6.43432 -12.14567
 -13.43221 -4.43212 -1.22456 0.43125 1.23098 4.43012 4.52567 5.4321 -4.43212 0.22132 1.42134 3.92041 -12.14567 5.12435 6.43432
```

```
-13.43221 -4.43212 -1.22456 0.43125 1.23098 4.43012 4.52567 5.4321 -12.14567 -4.43212 0.22132 1.42134 3.92041 5.12435 6.43432
-13.43221 -4.43212 -1.22456 0.43125 1.23098 4.43012 4.52567 5.4321 -12.14567 -4.43212 0.22132 1.42134 3.92041 5.12435 6.43432
-13.43221 -12.14567 -4.43212 -4.43212 -1.22456 0.22132 0.43125 1.23098 1.42134 3.92041 4.43012 4.52567 5.12435 5.4321 6.43432
After sorting:  -13.43221 -12.14567 -4.43212 -4.43212 -1.22456 0.22132 0.43125 1.23098 1.42134 3.92041 4.43012 4.52567 5.12435 5.4321 6.43432
```



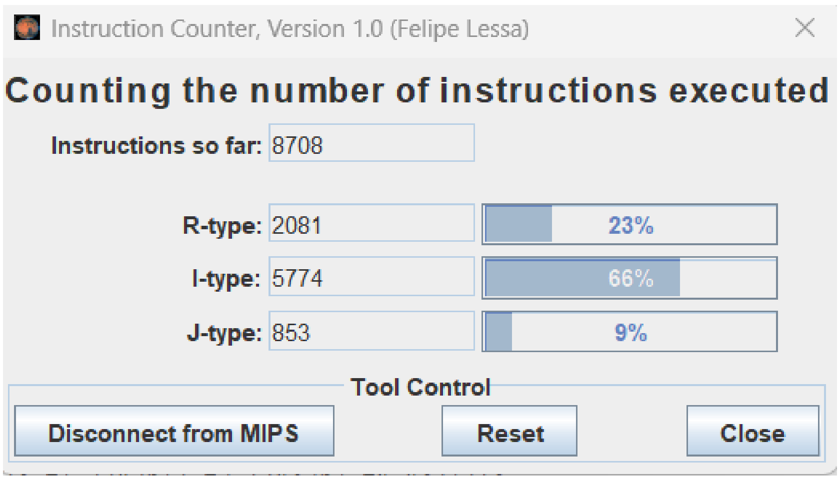
$$\text{CPU time} = \frac{\textit{Instruction Count} \times \textit{CPI}}{\textit{Clock rate}} = \frac{8788 \times 1}{1 \times 10^9} = 8.788us$$

Testcase 4: -4.43012345, -0.43125432, -5.43218765, -3.92041234, -13.43219876, -1.23098765, -1.42134567, -4.53201234, -12.14567890, -

6.43432123, -4.52567890, -0.22132123, -1.22456123, -5.12435678, -4.43212123

```
Before sorting:  -4.4301233 -0.43125433 -5.4321876 -3.9204123 -13.432199 -1.2309877 -1.4213457 -4.5320125 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -4.4301233 -0.43125433 -5.4321876 -3.9204123 -13.432199 -1.2309877 -1.4213457 -4.5320125 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -4.4301233 -0.43125433 -5.4321876 -3.9204123 -13.432199 -1.2309877 -1.4213457 -4.5320125 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -4.4301233 -0.43125433 -5.4321876 -3.9204123 -13.432199 -1.2309877 -1.4213457 -4.5320125 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -4.4301233 -0.43125433 -5.4321876 -3.9204123 -13.432199 -1.2309877 -1.4213457 -4.5320125 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -4.4301233 -0.43125433 -5.4321876 -3.9204123 -13.432199 -1.2309877 -1.4213457 -4.5320125 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -4.4301233 -0.43125433 -5.4321876 -3.9204123 -13.432199 -1.2309877 -1.4213457 -4.5320125 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -4.4301233 -0.43125433 -5.4321876 -3.9204123 -13.432199 -1.2309877 -1.4213457 -4.5320125 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -4.4301233 -0.43125433 -5.4321876 -3.9204123 -13.432199 -1.2309877 -1.4213457 -4.5320125 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -5.4321876 -4.4301233 -3.9204123 -0.43125433 -13.432199 -1.2309877 -1.4213457 -4.5320125 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -5.4321876 -4.4301233 -3.9204123 -0.43125433 -13.432199 -1.2309877 -1.4213457 -4.5320125 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -5.4321876 -4.4301233 -3.9204123 -0.43125433 -13.432199 -1.2309877 -1.4213457 -4.5320125 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -5.4321876 -4.4301233 -3.9204123 -0.43125433 -13.432199 -1.2309877 -1.4213457 -4.5320125 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -5.4321876 -4.4301233 -3.9204123 -0.43125433 -13.432199 -1.2309877 -1.4213457 -4.5320125 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -5.4321876 -4.4301233 -3.9204123 -0.43125433 -13.432199 -1.2309877 -1.4213457 -4.5320125 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -5.4321876 -4.4301233 -3.9204123 -0.43125433 -13.432199 -1.2309877 -1.4213457 -4.5320125 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -5.4321876 -4.4301233 -3.9204123 -0.43125433 -13.432199 -1.2309877 -1.4213457 -4.5320125 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -5.4321876 -4.4301233 -3.9204123 -0.43125433 -13.432199 -1.2309877 -4.5320125 -1.4213457 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -5.4321876 -4.4301233 -3.9204123 -0.43125433 -13.432199 -1.2309877 -4.5320125 -1.4213457 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -5.4321876 -4.4301233 -3.9204123 -0.43125433 -13.432199 -4.5320125 -1.4213457 -1.2309877 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -5.4321876 -4.4301233 -3.9204123 -0.43125433 -13.432199 -4.5320125 -1.4213457 -1.2309877 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -13.432199 -5.4321876 -4.5320125 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -0.43125433 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -13.432199 -5.4321876 -4.5320125 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -0.43125433 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -13.432199 -5.4321876 -4.5320125 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -0.43125433 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -13.432199 -5.4321876 -4.5320125 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -0.43125433 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -13.432199 -5.4321876 -4.5320125 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -0.43125433 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -13.432199 -5.4321876 -4.5320125 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -0.43125433 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -13.432199 -5.4321876 -4.5320125 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -0.43125433 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -13.432199 -5.4321876 -4.5320125 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -0.43125433 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -13.432199 -5.4321876 -4.5320125 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -0.43125433 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -13.432199 -5.4321876 -4.5320125 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -0.43125433 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -13.432199 -5.4321876 -4.5320125 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -0.43125433 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -13.432199 -5.4321876 -4.5320125 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -0.43125433 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -13.432199 -5.4321876 -4.5320125 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -0.43125433 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -13.432199 -5.4321876 -4.5320125 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -0.43125433 -12.1456785 -6.4343214 -4.525679 -0.22132123 -1.2245612 -5.1243567 -4.4321213
 -13.432199 -5.4321876 -4.5320125 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -0.43125433 -12.1456785 -6.4343214 -4.525679 -0.22132123 -5.1243567 -1.2245612 -4.4321213
 -13.432199 -5.4321876 -4.5320125 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -0.43125433 -12.1456785 -6.4343214 -4.525679 -0.22132123 -5.1243567 -1.2245612 -4.4321213
 -13.432199 -5.4321876 -4.5320125 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -0.43125433 -12.1456785 -6.4343214 -4.525679 -0.22132123 -5.1243567 -1.2245612 -4.4321213
 -13.432199 -5.4321876 -4.5320125 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -0.43125433 -12.1456785 -6.4343214 -4.525679 -0.22132123 -5.1243567 -4.4321213 -1.2245612


-13.432199 -5.4321876 -4.5320125 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -0.43125433 -12.1456785 -6.4343214 -4.525679 -0.22132123 -5.1243567 -4.4321213 -1.2245612
-13.432199 -5.4321876 -4.5320125 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -0.43125433 -12.1456785 -6.4343214 -5.1243567 -4.525679 -4.4321213 -1.2245612 -0.22132123
-13.432199 -5.4321876 -4.5320125 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -0.43125433 -12.1456785 -6.4343214 -5.1243567 -4.525679 -4.4321213 -1.2245612 -0.22132123
-13.432199 -12.1456785 -6.4343214 -5.4321876 -5.1243567 -4.5320125 -4.525679 -4.4321213 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -1.2245612 -0.43125433 -0.22132123
After sorting:  -13.432199 -12.1456785 -6.4343214 -5.4321876 -5.1243567 -4.5320125 -4.525679 -4.4321213 -4.4301233 -3.9204123 -1.4213457 -1.2309877 -1.2245612 -0.43125433 -0.22132123
```



$$\text{CPU time} = \frac{\textit{Instruction Count} \times CPI}{\textit{Clock rate}} = \frac{8708 \times 1}{1 \times 10^9} = 8708us$$

VII. Tổng kết:

6.1 Nhận xét sự hiệu quả của giải thuật Nhận xét:

- Là một thuật toán sắp xếp hiệu quả, thường được sử dụng trong các chương trình có khối lượng thao tác sắp xếp lớn.
- Độ phức tạp thuật toán là O(nlog(n))
- Không gian bộ nhớ sử dụng: O(n) , đem lại lợi thế về mặt bộ nhớ lưu trữ.
- Có tính ổn định(không làm thay đổi trình tự xuất hiện các phần tử giống nhau trong mảng)

6.2 Bảng phân chia công việc mỗi thành viên trong nhóm

Sinh viên thực hiện	Mã số sinh viên	Công việc
Nguyễn Nhật Tân	2213059	Code merge sort
Đào Duy Thành	2112288	Code tạo file BIN, làm báo cáo

6.3 Thu hoạch sau bài tập lớn

- Biết thêm được một giải thuật sắp xếp Merge Sort, thu thập được những kiến thức xung quanh giải thuật, viết được chương trình hợp ngữ đơn giản.
- Trau dồi kĩ năng làm việc nhóm cũng như việc lập trình hợp ngữ