

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



## KIẾN TRÚC MÁY TÍNH (CO2007)

---

### Bài tập lớn Kiến trúc tập lệnh MIPS

---

GVHD: Trần Thanh Bình  
SV thực hiện: Nguyễn Thị Duyên Bằng – 1912705 – L03  
Trịnh Duy Hưng – 1913652 – L03  
Nguyễn Hoàng Lâm – 1913918 – L03

Tp. Hồ Chí Minh, tháng 12/2020



## Mục lục

<b>1</b>	<b>Đề bài và yêu cầu</b>	<b>2</b>
1.1	Yêu cầu . . . . .	2
1.2	Đề bài . . . . .	2
<b>2</b>	<b>Ý tưởng của thuật toán merge sort</b>	<b>2</b>
<b>3</b>	<b>Code và mô tả phần code</b>	<b>2</b>
3.1	Code hợp ngữ MIPS . . . . .	2
3.2	Kết quả chạy trên MARS . . . . .	6
<b>4</b>	<b>Thống kê số lệnh, loại lệnh được thực thi của chương trình</b>	<b>13</b>
<b>5</b>	<b>Thời gian chạy của chương trình</b>	<b>14</b>
5.1	Công thức tính . . . . .	14
5.2	Thời gian thực thi . . . . .	14

# 1 Đề bài và yêu cầu

## 1.1 Yêu cầu

- Mỗi nhóm sinh viên làm một đề.
- Sử dụng tập lệnh MIPS để thực hiện các thủ tục bên dưới.
- Thống kê số lệnh, loại lệnh của chương trình của nhóm.
- Tính và trình bày cách tính thời gian chạy của chương trình trên máy tính MIPS có tần số 2GHz.
- Code:
  - Code style phải rõ ràng, có comment, phân hoạch công việc theo từng hàm.
  - Truyền nhận và trả kết quả gọi hàm theo quy ước sử dụng thanh ghi (\$a0 \$a3 cho argument, \$v0 v1 cho kết quả trả về).
  - Xuất kết quả để kiểm tra (sử dụng các hàm hệ thống).

## 1.2 Đề bài

### Đề 6

- Sắp xếp chuỗi.
- Cho một chuỗi số nguyên 20 phần tử. Sử dụng hợp ngữ assembly MIPS, viết thủ tục sắp xếp chuỗi đó theo tứ tự tăng dần theo giải thuật merge sort. Yêu cầu xuất ra từng bước trong quá trình demo.

# 2 Ý tưởng của thuật toán merge sort

Merge sort là một thuật toán chia để trị. Thuật toán này chia mảng cần sắp xếp thành 2 nửa. Tiếp tục lặp lại việc này ở các nửa mảng đã chia. Sau cùng gộp các nửa đó thành mảng đã sắp xếp. Hàm merge() được sử dụng để gộp hai nửa mảng. Hàm merge(arr, l, m, r) là tiến trình quan trọng nhất sẽ gộp hai nửa mảng thành 1 mảng sắp xếp, các nửa mảng là  $arr[l \dots m]$  và  $arr[m + 1 \dots r]$  sau khi gộp sẽ thành một mảng duy nhất đã sắp xếp.

# 3 Code và mô tả phần code

## 3.1 Code hợp ngữ MIPS

Một chương trình MIPS bao gồm 2 phần:

- .data: Khai báo các biến cần dùng sau chỉ thị này.
- .text: Viết các lệnh sau chỉ thị này.

a) Khai báo biến

- Khai báo chuỗi số nguyên gồm 20 phần tử
- Chuẩn bị tham số truyền vào hàm merge sort
- Thanh ghi a0 sẽ giữ địa chỉ con trỏ đầu tiên của mảng
- Thanh ghi a1 sẽ giữ địa chỉ con trỏ cuối cùng của mảng

```
# prepare parameter
la      $a0, my_array      # a0 giữ vị trí element đầu tiên của array

lw      $t0, array_length  # t0 = length
mul     $t0, $t0, 4         # t0 = size của array
add     $a1, $a0, $t0      # a1 giữ vị trí element cuối cùng của array

#-----
jal      Merge_sort        # Merge sort

#-----
jal      print_sorted_array # print sorted array
```

b) Thực hiện các lệnh và gọi hàm trong chương trình

- Gọi hàm merge sort.
- Tạo 4 chỗ trống để lưu lại những số cần thiết để sử dụng lại sau khi kết thúc đệ quy.
- Thanh ghi t0 sẽ chứa kích cỡ mảng hiện tại (số phần tử \* 4) bằng cách ta lấy địa chỉ phần tử cuối trừ địa chỉ phần tử đầu.
- Nếu mảng ta đang xét chỉ chứa 1 phần tử (size = 4) thì ta kết thúc để bước sang đệ quy mảng kế tiếp
- Về phần tính giá trị mid, ta phải thực hiện chia cho 8 sau đó nhân 4 để tránh việc bị sai địa chỉ (dẫn tới lỗi truy xuất giá trị phía sau code).

```
Merge_sort:
#          STACK
#          -----
#          | mid |
#          | end |
#          | start|
#          | ra  |
#          -----
addi      $sp, $sp, -16      # tạo 4 empty trong stack
sw        $ra, 0($sp)       # lưu ra vào stack để return sau
sw        $a0, 4($sp)       # lưu start vào stack
sw        $a1, 8($sp)       # lưu end vào stack

sub       $t0, $a1, $a0     # t0 = size of current array
                        # number of elements * 4

sle       $t9, $t0, 4       # Nếu tạo sort chỉ còn 1 phần tử thì ngưng
beq       $t9, 1, Stop_sort # t0 = size = 4

div       $t0, $t0, 8        # t0 = size / 8
mul       $t0, $t0, 4        # t0 = t0 * 4
add       $a1, $a0, $t0     # a1 = a0 + t0 = mid of current array
sw        $a1, 12($sp)      # lưu mid vào stack
```

- Lấy ra giá trị đã được lưu trong stack để truyền vào thanh ghi a0, a1, sau đó tiến hành đệ quy trên mảng liên kế tiếp.

```

# Recursive on first half
jal    Merge_sort

# After all first-half, return here

lw     $a0, 12($sp)      # new start = mid
lw     $a1, 8($sp)       # old end

# Recursive on second half
jal    Merge_sort

# After all first-half, return here

lw     $a0, 4($sp)       # restore Start
lw     $a1, 12($sp)      # restore Mid
lw     $a2, 8($sp)       # restore End

```

- Sau khi đã tách ra được 2 mảng con, tiến hành so sánh và ghép 2 mảng.

```

jal    Merge_array      # tien hanh ghep 2 mang lai

```

- Tạo 4 chỗ trống trong stack để giữ lại các giá trị cần thiết.
- Trước khi bước vào vòng lặp, cần tạo ra 2 bản sao của 2 mảng con, nhằm mục đích tránh sai sót.

```

Merge_array:
addi    $sp, $sp, -16      # tao 4 empty trong stack
sw      $ra, 0($sp)        # luu ra vao stack
sw      $a0, 4($sp)        # luu start vao stack
sw      $a1, 8($sp)        # luu mid vao stack
sw      $a2, 12($sp)       # luu end vao stack

move    $s0, $a0           # Tao 1 ban sao cua mang thu nhât
move    $s1, $a1           # Tao 1 ban sao cua mang thu hai

```

- Lần lượt tìm đến con trỏ tại vị trí cần thực hiện, lấy giá trị mà 2 con trỏ đó đang giữ.
- Nếu như giá trị con trỏ 1 (đứng trước trong danh sách) nhỏ hơn giá trị con trỏ 2 (đứng sau trong danh sách) thì đó đang là thứ tự ta mong muốn, nếu vậy ta sẽ không hoán đổi 2 giá trị.

```

Merge_loop:
lw      $t0, 0($s0)
lw      $t1, 0($s1)

lw      $t0, 0($t0)
lw      $t1, 0($t1)

blt     $t0, $t1, No_swap

```

- Hàm no - swap
- Nếu như không xảy ra hoán đổi, ta tăng con trỏ tại mảng thứ nhất lên vị trí tiếp theo (con trỏ tại mảng thứ hai vẫn giữ nguyên) để tiếp tục so sánh hết mảng.
- Lấy ra từ stack giá trị end của mảng hiện tại và so sánh, nếu như một trong hai con trỏ tại hai mảng gặp vị trí cuối thì ta kết thúc hàm merge array.

**No\_swap:**

```
addi    $s0, $s0, 4      # Tang ptr o mang
lw      $a2, 12($sp)     # restore end

bge     $s1, $a2, Merge_end # Ket thuc neu nhu
bge     $s0, $a2, Merge_end

# tiep tục so sánh nếu chưa duyệt hết
j       Merge_loop
```

- Nếu như giá trị đứng trước lớn hơn giá trị đứng sau, ta cần phải hoán đổi nó.
- Truyền vào a0, a1 hai giá trị và bước vào hàm swap để hoán đổi.
- Sau đó, ta sẽ in ra mảng sau khi swap. Tăng con trỏ tại mảng thứ hai đến vị trí kế tiếp.

```
# swap
move     $a0, $s1

move     $a1, $s0

jal      Swap
jal      print_current_array

addi     $s1, $s1, 4
```

- Hàm swap
- Nếu như hai con trỏ trùng nhau, t9 sẽ là 1 và ta kết thúc hàm swap.
- a0 hiện tại là con trỏ ở mảng thứ hai, nó sẽ lui dần từng phần tử một.
- a1 hiện tại sẽ giữ vị trí đích đến tại mảng 1.
- s5 sẽ là phần tử đứng liền trước a0.
- Hoán đổi chúng.
- Như vậy khi gặp nhau a1 sẽ giữ giá trị đúng của nó.
- Hàm swap này sẽ được gọi nhiều lần cho đến khi tất cả các phần tử giữa chúng được xếp đúng thứ tự.

Swap:

```
sle    $t9, $a0, $a1      # neu nhu hai dia chi da trung nhau, stop swapping
beq    $t9, 1, Swap_end

sub    $s5, $a0, 4        # Lui ve 1 index

lw     $s6, 0($a0)        # lay ptr index a
lw     $s7, 0($s5)        # lay ptr index a - 1

sw     $s6, 0($s5)        # hona doi gia tri trong ptr
sw     $s7, 0($a0)        #

sub    $a0, $a0, 4        # current position -= 1
j      Swap               # Tiep tục vòng lặp

Swap_end:
jr     $ra                # Return
```

- Lấy lại return address để trở về, xóa stack.

```
Merge_end:
lw     $ra, 0($sp)        # restore ra
addi   $sp, $sp, 16       # xia empty da
jr     $ra                # Return
Swap an array element to another position, at a lower address
Thay đổi một phần tử vào đúng vị trí cần swap
```

## 3.2 Kết quả chạy trên MARS

1. Testcase 1 [10, -5, 2, -6, -2, 1, 7, 3, -3, -1, 6, 0, -4, 5, -8, -7, 8, -9, 9, 4]

```
Unsorted Array: [10, -5, 2, -6, -2, 1, 7, 3, -3, -1, 6, 0, -4, 5, -8, -7, 8, -9, 9, 4]
-----
Current array:  [-5, 10, 2, -6, -2, 1, 7, 3, -3, -1, 6, 0, -4, 5, -8, -7, 8, -9, 9, 4]
-----
Current array:  [-5, 10, 2, -6, -2, 1, 7, 3, -3, -1, 6, 0, -4, 5, -8, -7, 8, -9, 9, 4]
-----
Current array:  [-5, 10, -6, 2, -2, 1, 7, 3, -3, -1, 6, 0, -4, 5, -8, -7, 8, -9, 9, 4]
-----
Current array:  [-5, 10, -6, -2, 2, 1, 7, 3, -3, -1, 6, 0, -4, 5, -8, -7, 8, -9, 9, 4]
-----
Current array:  [-6, -5, 10, -2, 2, 1, 7, 3, -3, -1, 6, 0, -4, 5, -8, -7, 8, -9, 9, 4]
-----
Current array:  [-6, -5, -2, 10, 2, 1, 7, 3, -3, -1, 6, 0, -4, 5, -8, -7, 8, -9, 9, 4]
-----
Current array:  [-6, -5, -2, 2, 10, 1, 7, 3, -3, -1, 6, 0, -4, 5, -8, -7, 8, -9, 9, 4]
-----
Current array:  [-6, -5, -2, 2, 10, 1, 7, 3, -3, -1, 6, 0, -4, 5, -8, -7, 8, -9, 9, 4]
-----
Current array:  [-6, -5, -2, 2, 10, 1, 7, -3, 3, -1, 6, 0, -4, 5, -8, -7, 8, -9, 9, 4]
-----
Current array:  [-6, -5, -2, 2, 10, 1, 7, -3, -1, 3, 6, 0, -4, 5, -8, -7, 8, -9, 9, 4]
-----
Current array:  [-6, -5, -2, 2, 10, -3, 1, 7, -1, 3, 6, 0, -4, 5, -8, -7, 8, -9, 9, 4]
-----
Current array:  [-6, -5, -2, 2, 10, -3, -1, 1, 7, 3, 6, 0, -4, 5, -8, -7, 8, -9, 9, 4]
-----
Current array:  [-6, -5, -2, 2, 10, -3, -1, 1, 3, 7, 6, 0, -4, 5, -8, -7, 8, -9, 9, 4]
-----
```

```
Current array: [-6, -5, -3, -2, 2, 10, -1, 1, 3, 7, 6, 0, -4, 5, -8, -7, 8, -9, 9, 4]
-----
Current array: [-6, -5, -3, -2, -1, 2, 10, 1, 3, 7, 6, 0, -4, 5, -8, -7, 8, -9, 9, 4]
-----
Current array: [-6, -5, -3, -2, -1, 1, 2, 10, 3, 7, 6, 0, -4, 5, -8, -7, 8, -9, 9, 4]
-----
Current array: [-6, -5, -3, -2, -1, 1, 2, 3, 10, 7, 6, 0, -4, 5, -8, -7, 8, -9, 9, 4]
-----
Current array: [-6, -5, -3, -2, -1, 1, 2, 3, 7, 10, 6, 0, -4, 5, -8, -7, 8, -9, 9, 4]
-----
Current array: [-6, -5, -3, -2, -1, 1, 2, 3, 7, 10, 0, 6, -4, 5, -8, -7, 8, -9, 9, 4]
-----
Current array: [-6, -5, -3, -2, -1, 1, 2, 3, 7, 10, 0, 6, -4, -8, 5, -7, 8, -9, 9, 4]
-----
Current array: [-6, -5, -3, -2, -1, 1, 2, 3, 7, 10, 0, 6, -8, -4, 5, -7, 8, -9, 9, 4]
-----
Current array: [-6, -5, -3, -2, -1, 1, 2, 3, 7, 10, 0, 6, -8, -4, 5, -7, 8, -9, 9, 4]
-----
Current array: [-6, -5, -3, -2, -1, 1, 2, 3, 7, 10, -8, 0, 6, -4, 5, -7, 8, -9, 9, 4]
-----
Current array: [-6, -5, -3, -2, -1, 1, 2, 3, 7, 10, -8, -4, 0, 6, 5, -7, 8, -9, 9, 4]
-----
Current array: [-6, -5, -3, -2, -1, 1, 2, 3, 7, 10, -8, -4, 0, 5, 6, -7, 8, -9, 9, 4]

Current array: [-6, -5, -3, -2, -1, 1, 2, 3, 7, 10, -8, -4, 0, 5, 6, -7, 8, -9, 9, 4]
-----
Current array: [-6, -5, -3, -2, -1, 1, 2, 3, 7, 10, -8, -4, 0, 5, 6, -7, 8, -9, 4, 9]
-----
Current array: [-6, -5, -3, -2, -1, 1, 2, 3, 7, 10, -8, -4, 0, 5, 6, -7, 8, -9, 4, 9]
-----
Current array: [-6, -5, -3, -2, -1, 1, 2, 3, 7, 10, -8, -4, 0, 5, 6, -7, 8, -9, 4, 9]
-----
Current array: [-6, -5, -3, -2, -1, 1, 2, 3, 7, 10, -8, -4, 0, 5, 6, -9, -7, 8, 4, 9]
-----
Current array: [-6, -5, -3, -2, -1, 1, 2, 3, 7, 10, -8, -4, 0, 5, 6, -9, -7, 4, 8, 9]
-----
Current array: [-6, -5, -3, -2, -1, 1, 2, 3, 7, 10, -8, -4, 0, 5, 6, -9, -7, 4, 8, 9]
-----
Current array: [-6, -5, -3, -2, -1, 1, 2, 3, 7, 10, -9, -8, -4, 0, 5, 6, -7, 4, 8, 9]
-----
Current array: [-6, -5, -3, -2, -1, 1, 2, 3, 7, 10, -9, -8, -7, -4, 0, 5, 6, 4, 8, 9]
-----
Current array: [-6, -5, -3, -2, -1, 1, 2, 3, 7, 10, -9, -8, -7, -4, 0, 4, 5, 6, 8, 9]
-----
Current array: [-6, -5, -3, -2, -1, 1, 2, 3, 7, 10, -9, -8, -7, -4, 0, 4, 5, 6, 8, 9]
-----
Current array: [-6, -5, -3, -2, -1, 1, 2, 3, 7, 10, -9, -8, -7, -4, 0, 4, 5, 6, 8, 9]
```

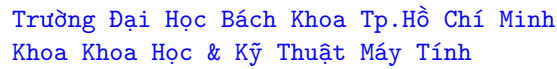


```
Current array: [-9, -8, -6, -5, -3, -2, -1, 1, 2, 3, 7, 10, -7, -4, 0, 4, 5, 6, 8, 9]
-----
Current array: [-9, -8, -7, -6, -5, -3, -2, -1, 1, 2, 3, 7, 10, -4, 0, 4, 5, 6, 8, 9]
-----
Current array: [-9, -8, -7, -6, -5, -4, -3, -2, -1, 1, 2, 3, 7, 10, 0, 4, 5, 6, 8, 9]
-----
Current array: [-9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 7, 10, 4, 5, 6, 8, 9]
-----
Current array: [-9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 7, 10, 5, 6, 8, 9]
-----
Current array: [-9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 7, 10, 6, 8, 9]
-----
Current array: [-9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 10, 8, 9]
-----
Current array: [-9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 9]
-----
Current array: [-9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
-----
Sorted Array:  [-9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Chuỗi sau khi sắp xếp: [-9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

## 2. Testcase 2 [12, 34, 5, -9, -23, 1, 6, 4, -10, -35, 10, 0, -4, 12, -11, -24, 24, -23, 1, 56]

```
Unsorted Array: [12, 34, 5, -9, -23, 1, 6, 4, -10, -35, 10, 0, -4, 12, -11, -24, 24, -23, 1, 56]
-----
Current array: [12, 34, 5, -9, -23, 1, 6, 4, -10, -35, 10, 0, -4, 12, -11, -24, 24, -23, 1, 56]
-----
Current array: [12, 34, 5, -23, -9, 1, 6, 4, -10, -35, 10, 0, -4, 12, -11, -24, 24, -23, 1, 56]
-----
Current array: [12, 34, -23, 5, -9, 1, 6, 4, -10, -35, 10, 0, -4, 12, -11, -24, 24, -23, 1, 56]
-----
Current array: [12, 34, -23, -9, 5, 1, 6, 4, -10, -35, 10, 0, -4, 12, -11, -24, 24, -23, 1, 56]
-----
Current array: [-23, 12, 34, -9, 5, 1, 6, 4, -10, -35, 10, 0, -4, 12, -11, -24, 24, -23, 1, 56]
-----
Current array: [-23, -9, 12, 34, 5, 1, 6, 4, -10, -35, 10, 0, -4, 12, -11, -24, 24, -23, 1, 56]
-----
Current array: [-23, -9, 5, 12, 34, 1, 6, 4, -10, -35, 10, 0, -4, 12, -11, -24, 24, -23, 1, 56]
-----
Current array: [-23, -9, 5, 12, 34, 1, 6, 4, -10, -35, 10, 0, -4, 12, -11, -24, 24, -23, 1, 56]
-----
Current array: [-23, -9, 5, 12, 34, 1, 6, 4, -35, -10, 10, 0, -4, 12, -11, -24, 24, -23, 1, 56]
-----
Current array: [-23, -9, 5, 12, 34, 1, 6, -35, 4, -10, 10, 0, -4, 12, -11, -24, 24, -23, 1, 56]
-----
Current array: [-23, -9, 5, 12, 34, 1, 6, -35, -10, 4, 10, 0, -4, 12, -11, -24, 24, -23, 1, 56]
-----
Current array: [-23, -9, 5, 12, 34, -35, 1, 6, -10, 4, 10, 0, -4, 12, -11, -24, 24, -23, 1, 56]
-----
Current array: [-23, -9, 5, 12, 34, -35, -10, 1, 6, 4, 10, 0, -4, 12, -11, -24, 24, -23, 1, 56]
-----
Current array: [-23, -9, 5, 12, 34, -35, -10, 1, 4, 6, 10, 0, -4, 12, -11, -24, 24, -23, 1, 56]
-----
Current array: [-35, -23, -9, 5, 12, 34, -10, 1, 4, 6, 10, 0, -4, 12, -11, -24, 24, -23, 1, 56]
```



Chuỗi sau khi sắp xếp:  $[-35, -24, -23, -23, -11, -10, -9, -4, 0, 1, 1, 4, 5, 6, 10, 12, 12, 24, 34, 56]$

Các testcase tương tự (hình ảnh chi tiết được đính kèm trong thư mục testcase)

3. Testcase 3 [25, 72, 47, -62, -42, 42, 32, 14, -12, -14, 34, 0, -47, 87, -99, -100, 21, -53, 1, 72]

```
Unsorted Array: [25, 72, 47, -62, -42, 42, 32, 14, -12, -14, 34, 0, -47, 87, -99, -100, 21, -53, 1, 72]
-----
Current array: [25, 72, 47, -62, -42, 42, 32, 14, -12, -14, 34, 0, -47, 87, -99, -100, 21, -53, 1, 72]
-----
Current array: [25, 72, 47, -62, -42, 42, 32, 14, -12, -14, 34, 0, -47, 87, -99, -100, 21, -53, 1, 72]
-----
Current array: [25, 72, -62, 47, -42, 42, 32, 14, -12, -14, 34, 0, -47, 87, -99, -100, 21, -53, 1, 72]
-----
...
Current array: [-100, -99, -62, -53, -47, -42, -14, -12, 0, 1, 14, 21, 25, 32, 34, 42, 47, 72, 72, 87]
-----
Current array: [-100, -99, -62, -53, -47, -42, -14, -12, 0, 1, 14, 21, 25, 32, 34, 42, 47, 72, 72, 87]
-----
Sorted Array: [-100, -99, -62, -53, -47, -42, -14, -12, 0, 1, 14, 21, 25, 32, 34, 42, 47, 72, 72, 87]
```

Chuỗi sau khi sắp xếp: [-100, -99, -62, -53, -47, -42, -14, -12, 0, 1, 14, 21, 25, 32, 34, 42, 47, 72, 72, 87]

4. Testcase 4 [41, 25, 75, -6, -4, 87, 37, 19, -10, -17, 85, 0, -47, 87, -63, -100, 27, -59, 1, 72]

```
Unsorted Array: [41, 25, 75, -6, -4, 87, 37, 19, -10, -17, 85, 0, -47, 87, -63, -100, 27, -59, 1, 72]
-----
Current array: [25, 41, 75, -6, -4, 87, 37, 19, -10, -17, 85, 0, -47, 87, -63, -100, 27, -59, 1, 72]
-----
Current array: [25, 41, 75, -6, -4, 87, 37, 19, -10, -17, 85, 0, -47, 87, -63, -100, 27, -59, 1, 72]
-----
Current array: [25, 41, -6, 75, -4, 87, 37, 19, -10, -17, 85, 0, -47, 87, -63, -100, 27, -59, 1, 72]
-----
...
Current array: [-100, -63, -59, -47, -17, -10, -6, -4, 0, 1, 19, 25, 27, 37, 41, 75, 87, 72, 85, 87]
-----
Current array: [-100, -63, -59, -47, -17, -10, -6, -4, 0, 1, 19, 25, 27, 37, 41, 72, 75, 87, 85, 87]
-----
Current array: [-100, -63, -59, -47, -17, -10, -6, -4, 0, 1, 19, 25, 27, 37, 41, 72, 75, 85, 87, 87]
-----
Current array: [-100, -63, -59, -47, -17, -10, -6, -4, 0, 1, 19, 25, 27, 37, 41, 72, 75, 85, 87, 87]
-----
Sorted Array: [-100, -63, -59, -47, -17, -10, -6, -4, 0, 1, 19, 25, 27, 37, 41, 72, 75, 85, 87, 87]
```

Chuỗi sau khi sắp xếp: [-100, -63, -59, -47, -17, -10, -6, -4, 0, 1, 19, 25, 27, 37, 41, 72, 75, 85, 87, 87]

5. Testcase 5 [123, 132, -876, -421, -541, 56, 156, 543, -588, -817, 643, 243, 0, 546, -879, -54, 676, -46, 999, 723]

```
Unsorted Array: [123, 132, -876, -421, -541, 56, 156, 543, -588, -817, 643, 243, 0, 546, -879, -54, 676, -46, 999, 723]
-----
Current array: [123, 132, -876, -421, -541, 56, 156, 543, -588, -817, 643, 243, 0, 546, -879, -54, 676, -46, 999, 723]
-----
Current array: [123, 132, -876, -541, -421, 56, 156, 543, -588, -817, 643, 243, 0, 546, -879, -54, 676, -46, 999, 723]
-----
Current array: [123, 132, -876, -541, -421, 56, 156, 543, -588, -817, 643, 243, 0, 546, -879, -54, 676, -46, 999, 723]
-----
Current array: [123, 132, -876, -541, -421, 56, 156, 543, -588, -817, 643, 243, 0, 546, -879, -54, 676, -46, 999, 723]
-----
...
Current array: [-879, -876, -817, -588, -541, -421, -54, -46, 0, 56, 123, 132, 156, 243, 543, 546, 643, 676, 723, 999]
-----
Current array: [-879, -876, -817, -588, -541, -421, -54, -46, 0, 56, 123, 132, 156, 243, 543, 546, 643, 676, 723, 999]
-----
Current array: [-879, -876, -817, -588, -541, -421, -54, -46, 0, 56, 123, 132, 156, 243, 543, 546, 643, 676, 723, 999]
-----
Current array: [-879, -876, -817, -588, -541, -421, -54, -46, 0, 56, 123, 132, 156, 243, 543, 546, 643, 676, 723, 999]
-----
Sorted Array: [-879, -876, -817, -588, -541, -421, -54, -46, 0, 56, 123, 132, 156, 243, 543, 546, 643, 676, 723, 999]
```

Chuỗi sau khi sắp xếp: [-879, -876, -817, -588, -541, -421, -54, -46, 0, 56, 123, 312, 156, 243, 543, 546, 643, 676, 723, 999]

6. Testcase 6 [12, 13, -87, -42, -54, 5, 15, 54, -58, -81, 64, 23, 0, 54, -79, -5, 67, -4, 99, 73]

```
Unsorted Array: [12, 13, -87, -42, -54, 5, 15, 54, -58, -81, 64, 23, 0, 54, -79, -5, 67, -4, 99, 73]
-----
Current array: [12, 13, -87, -42, -54, 5, 15, 54, -58, -81, 64, 23, 0, 54, -79, -5, 67, -4, 99, 73]
-----
Current array: [12, 13, -87, -54, -42, 5, 15, 54, -58, -81, 64, 23, 0, 54, -79, -5, 67, -4, 99, 73]
-----
Current array: [12, 13, -87, -54, -42, 5, 15, 54, -58, -81, 64, 23, 0, 54, -79, -5, 67, -4, 99, 73]
```

...

```
Current array: [-87, -81, -79, -58, -54, -42, -5, -4, 0, 5, 12, 13, 15, 23, 54, 54, 64, 67, 73, 99]
-----
Current array: [-87, -81, -79, -58, -54, -42, -5, -4, 0, 5, 12, 13, 15, 23, 54, 54, 64, 67, 73, 99]
-----
Current array: [-87, -81, -79, -58, -54, -42, -5, -4, 0, 5, 12, 13, 15, 23, 54, 54, 64, 67, 73, 99]
-----
Current array: [-87, -81, -79, -58, -54, -42, -5, -4, 0, 5, 12, 13, 15, 23, 54, 54, 64, 67, 73, 99]
-----
Sorted Array: [-87, -81, -79, -58, -54, -42, -5, -4, 0, 5, 12, 13, 15, 23, 54, 54, 64, 67, 73, 99]
```

Chuỗi sau khi sắp xếp: [-87, -81, -79, -58, -54, -42, -5, -4, 0, 5, 12, 13, 15, 23, 54, 54, 64, 67, 73, 99]

7. Testcase 7 [155, 843, -652, -673, -354, 62, 15, 565, -512, -875, 651, 476, 46, 0, -258, -546, -565, -54, 578, 735]

```
Unsorted Array: [155, 843, -652, -673, -354, 62, 15, 565, -512, -875, 651, 476, 46, 0, -258, -546, 565, -54, 578, 735]
-----
Current array: [155, 843, -652, -673, -354, 62, 15, 565, -512, -875, 651, 476, 46, 0, -258, -546, 565, -54, 578, 735]
-----
Current array: [155, 843, -652, -673, -354, 62, 15, 565, -512, -875, 651, 476, 46, 0, -258, -546, 565, -54, 578, 735]
-----
Current array: [155, 843, -673, -652, -354, 62, 15, 565, -512, -875, 651, 476, 46, 0, -258, -546, 565, -54, 578, 735]
-----
Current array: [155, 843, -673, -652, -354, 62, 15, 565, -512, -875, 651, 476, 46, 0, -258, -546, 565, -54, 578, 735]
-----
Current array: [-673, 155, 843, -652, -354, 62, 15, 565, -512, -875, 651, 476, 46, 0, -258, -546, 565, -54, 578, 735]
-----
```

...

```
Current array: [-875, -673, -652, -546, -512, -354, -258, -54, 0, 15, 46, 62, 155, 476, 565, 843, 565, 578, 651, 735]
-----
Current array: [-875, -673, -652, -546, -512, -354, -258, -54, 0, 15, 46, 62, 155, 476, 565, 565, 843, 578, 651, 735]
-----
Current array: [-875, -673, -652, -546, -512, -354, -258, -54, 0, 15, 46, 62, 155, 476, 565, 565, 578, 843, 651, 735]
-----
Current array: [-875, -673, -652, -546, -512, -354, -258, -54, 0, 15, 46, 62, 155, 476, 565, 565, 578, 651, 843, 735]
-----
Current array: [-875, -673, -652, -546, -512, -354, -258, -54, 0, 15, 46, 62, 155, 476, 565, 565, 578, 651, 735, 843]
-----
Sorted Array: [-875, -673, -652, -546, -512, -354, -258, -54, 0, 15, 46, 62, 155, 476, 565, 565, 578, 651, 735, 843]
```

Chuỗi sau khi sắp xếp: [-875, -673, -652, -546, -512, -354, -258, -54, 0, 15, 46, 62, 155, 476, 565, 565, 578, 651, 735, 843]

8. Testcase 8 [15, 84, -62, -63, -54, 612, 521, 56, -12, -75, 61, 47, 4, 0, -58, -46, 65, -54, 58, 75]

```
Unsorted Array: [15, 84, -62, -63, -54, 612, 521, 56, -12, -75, 61, 47, 4, 0, -58, -46, 65, -54, 58, 75]
-----
Current array: [15, 84, -62, -63, -54, 612, 521, 56, -12, -75, 61, 47, 4, 0, -58, -46, 65, -54, 58, 75]
-----
Current array: [15, 84, -62, -63, -54, 612, 521, 56, -12, -75, 61, 47, 4, 0, -58, -46, 65, -54, 58, 75]
-----
Current array: [15, 84, -63, -62, -54, 612, 521, 56, -12, -75, 61, 47, 4, 0, -58, -46, 65, -54, 58, 75]
-----
```



...

```
Current array: [-75, -63, -62, -58, -54, -54, -46, -12, 0, 4, 15, 47, 56, 58, 84, 521, 612, 61, 65, 75]
-----
Current array: [-75, -63, -62, -58, -54, -54, -46, -12, 0, 4, 15, 47, 56, 58, 61, 84, 521, 612, 65, 75]
-----
Current array: [-75, -63, -62, -58, -54, -54, -46, -12, 0, 4, 15, 47, 56, 58, 61, 65, 84, 521, 612, 75]
-----
Current array: [-75, -63, -62, -58, -54, -54, -46, -12, 0, 4, 15, 47, 56, 58, 61, 65, 75, 84, 521, 612]
-----
Sorted Array: [-75, -63, -62, -58, -54, -54, -46, -12, 0, 4, 15, 47, 56, 58, 61, 65, 75, 84, 521, 612]
```

Chuỗi sau khi sắp xếp: [-75, -63, -62, -58, -54, -54, -46, -12, 0, 4, 15, 47, 56, 58, 61, 65, 75, 84, 521, 612]

9. Testcase 9 [687, 124, -321, -65, -565, 654, 921, 135, -658, -145, 454, 789, 452, 0, -311, -798, 897, -654, -315, 268]

```
Unsorted Array: [687, 124, -321, -65, -565, 654, 921, 135, -658, -145, 454, 789, 452, 0, -311, -798, 897, -654, 315, 268]
-----
Current array: [124, 687, -321, -65, -565, 654, 921, 135, -658, -145, 454, 789, 452, 0, -311, -798, 897, -654, 315, 268]
-----
Current array: [124, 687, -321, -565, -65, 654, 921, 135, -658, -145, 454, 789, 452, 0, -311, -798, 897, -654, 315, 268]
-----
Current array: [124, 687, -565, -321, -65, 654, 921, 135, -658, -145, 454, 789, 452, 0, -311, -798, 897, -654, 315, 268]
-----
Current array: [124, 687, -565, -321, -65, 654, 921, 135, -658, -145, 454, 789, 452, 0, -311, -798, 897, -654, 315, 268]
```

...

```
Current array: [-798, -658, -654, -565, -321, -311, -145, -65, 0, 124, 135, 268, 315, 452, 454, 654, 687, 921, 789, 897]
-----
Current array: [-798, -658, -654, -565, -321, -311, -145, -65, 0, 124, 135, 268, 315, 452, 454, 654, 687, 789, 921, 897]
-----
Current array: [-798, -658, -654, -565, -321, -311, -145, -65, 0, 124, 135, 268, 315, 452, 454, 654, 687, 789, 897, 921]
-----
Sorted Array: [-798, -658, -654, -565, -321, -311, -145, -65, 0, 124, 135, 268, 315, 452, 454, 654, 687, 789, 897, 921]
```

Chuỗi sau khi sắp xếp: [-798, -658, -654, -565, -321, -311, -145, -65, 0, 124, 135, 268, 315, 452, 454, 654, 687, 789, 897, 921]

10. Testcase 10 [6222, 1448, -21, -853, -5524, 648, 917, 1655, -6585, -165, 4561, 771, 461, 32, 0, -984, 6156, -6872, 6542, 5462]

```
Unsorted Array: [6222, 1448, -21, -853, -5524, 648, 917, 1655, -6585, -165, 4561, 771, 461, 32, 0, -984, 6156, -6872, 6542, 5462]
-----
Current array: [1448, 6222, -21, -853, -5524, 648, 917, 1655, -6585, -165, 4561, 771, 461, 32, 0, -984, 6156, -6872, 6542, 5462]
-----
Current array: [1448, 6222, -21, -5524, -853, 648, 917, 1655, -6585, -165, 4561, 771, 461, 32, 0, -984, 6156, -6872, 6542, 5462]
-----
Current array: [1448, 6222, -5524, -21, -853, 648, 917, 1655, -6585, -165, 4561, 771, 461, 32, 0, -984, 6156, -6872, 6542, 5462]
-----
Current array: [1448, 6222, -5524, -853, -21, 648, 917, 1655, -6585, -165, 4561, 771, 461, 32, 0, -984, 6156, -6872, 6542, 5462]
```

...

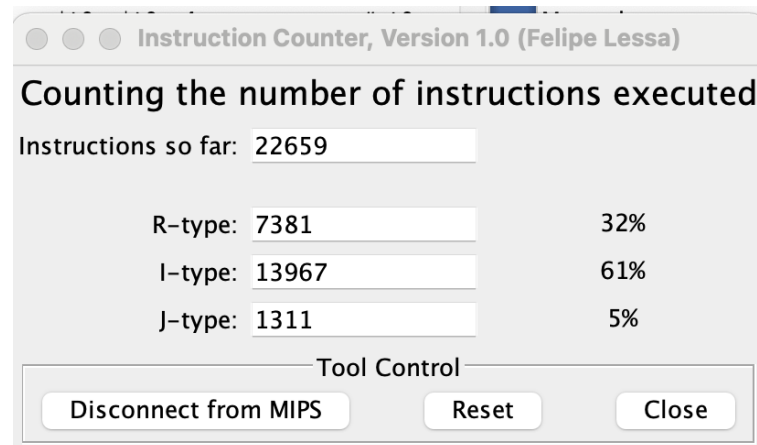
```
Current array: [-6872, -6585, -5524, -984, -853, -165, -21, 0, 32, 461, 648, 771, 917, 1448, 1655, 6222, 4561, 5462, 6156, 6542]
-----
Current array: [-6872, -6585, -5524, -984, -853, -165, -21, 0, 32, 461, 648, 771, 917, 1448, 1655, 4561, 6222, 5462, 6156, 6542]
-----
Current array: [-6872, -6585, -5524, -984, -853, -165, -21, 0, 32, 461, 648, 771, 917, 1448, 1655, 4561, 5462, 6222, 6156, 6542]
-----
Current array: [-6872, -6585, -5524, -984, -853, -165, -21, 0, 32, 461, 648, 771, 917, 1448, 1655, 4561, 5462, 6156, 6222, 6542]
-----
Current array: [-6872, -6585, -5524, -984, -853, -165, -21, 0, 32, 461, 648, 771, 917, 1448, 1655, 4561, 5462, 6156, 6222, 6542]
-----
Sorted Array: [-6872, -6585, -5524, -984, -853, -165, -21, 0, 32, 461, 648, 771, 917, 1448, 1655, 4561, 5462, 6156, 6222, 6542]
```

Chuỗi sau khi sắp xếp: [-6872, -6585, -5524, -984, -853, -165, -21, 0, 32, 461, 648, 771, 917, 1448, 1655, 4561, 5462, 6156, 6222, 6542]

## 4 Thống kê số lệnh, loại lệnh được thực thi của chương trình

Sử dụng plug-in (**Tools/Instruction Statistics**) trong MARS để thống kê số lượng lệnh và loại lệnh của chương trình.

- Ấn phím tắt **F3** hoặc vào menu **Run/Assemble**.
- Chọn **Tools/Instruction Counter**, chọn **Connect to MIPS**.
- Chạy chương trình.



Đối với chuỗi đã cho, thống kê số lệnh của mỗi loại lệnh như sau:

- R-type: 7381 lệnh. (32%)
- I-type: 13967 lệnh. (61%)
- J-type: 1311 lệnh. (5%)

## 5 Thời gian chạy của chương trình

### 5.1 Công thức tính

Cách tính thời gian chạy của chương trình trên máy tính MIPS có tần số 2Ghz:

$$\text{CPU time} = \text{Instruction Count} \times \text{CPI} / \text{Clock rate}$$

Trong đó:

- **CPU time:** Thời gian thực thi
- **Instruction Count:** Tổng số lệnh thực thi
- **CPI:** (clock cycle per instruction) Số chu kỳ cần để thực thi một lệnh = 1
- **Clock rate:** Tần số máy tính = 2 Ghz

### 5.2 Thời gian thực thi

Ví dụ:

Với testcase là chuỗi số: [10, -5, 2, -6, -2, 1, 7, 3, -3, -1, 6, 0, -4, 5, -8, -7, 8, -9, 9, 4]

Instruction Count = 22659

$$\begin{aligned}\text{CPU time} &= (\text{Instruction Count} \times \text{CPI}) / \text{Clock rate} \\ &= (22659 \times 1) / (2 \times 10^9) \\ &= 11.3295 \times 10^{-6} \text{ (s)}\end{aligned}$$