

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



BÁO CÁO BÀI TẬP LỚN
MÔN HỌC: KIẾN TRÚC MÁY TÍNH (CO2007)

ĐỀ TÀI:

SELECTION SORT SỐ NGUYÊN

LỚP L01 - NHÓM BTL059 - HK 242

Giảng viên hướng dẫn: NGUYỄN XUÂN MINH

TP Hồ Chí Minh, ngày 20 tháng 05 năm 2024



BÁO CÁO KẾT QUẢ LÀM VIỆC NHÓM

MSSV	Họ và tên	Tỷ lệ %	Điểm	Ký tên
2210147	Trương Nguyễn Hoàng Anh	100%		<i>H.Anh</i>
2210885	Lê Xuân Hải	100%		<i>X.Hải</i>

Nhận xét của giảng viên:

.....

.....

.....

.....

.....

.....

GIẢNG VIÊN

(Ký và ghi rõ họ, tên)

MỤC LỤC

I. Đề tài	3
II. Giải pháp hiện thực và giải thuật	3
1. Thuật toán Selection sort	3
2. Giải pháp hiện thực	5
a. Phân tích luồng thực thi :	5
b. Các hàm sử dụng để giải quyết bài toán	6
b. Thống kê số lệnh, kiểu lệnh	6
III. Chạy chương trình với các testcase	8
a. Một số testcase sử dụng :	8
b. Kết quả và thời gian thực thi của chương trình	8
IV. Cài đặt và chạy code	10

I. Đề tài :

Selection Sort số nguyên : Viết chương trình sắp xếp dãy số nguyên có 10 phần tử dùng giải thuật Selection sort. Yêu cầu xuất ra màn hình các bước có thay đổi thứ tự trong dãy. Dữ liệu đầu vào đọc từ file lưu trữ dạng nhị phân trên đĩa INT10.BIN (10 phần tử x 4 bytes = 40 bytes).

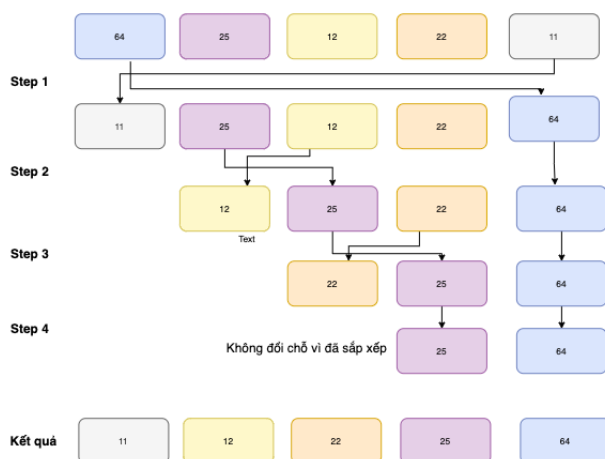
II. Giải pháp hiện thực và giải thuật :

1. Thuật toán Selection sort :

Thuật toán selection sort sắp xếp một mảng bằng cách đi tìm phần tử có giá trị nhỏ nhất(giả sử với sắp xếp mảng tăng dần) trong đoạn đoạn chưa được sắp xếp và đổi cho phần tử nhỏ nhất đó với phần tử ở đầu đoạn chưa được sắp xếp(không phải đầu mảng). Thuật toán sẽ chia mảng làm 2 mảng con :

1. Một mảng con đã được sắp xếp
2. Một mảng con chưa được sắp xếp

Tại mỗi bước lặp của thuật toán, phần tử nhỏ nhất ở mảng con chưa được sắp xếp sẽ được di chuyển về đoạn đã sắp xếp.



Hình 1: Ví dụ về thuật toán Selection Sort

Có thể giải thích ví dụ trên như sau :

```
1 //Day chua sap xep
2 arr[] = 64 25 12 22 11
3 // Tim phan tu be nhat trong day arr[0...4]
4 // dua phan tu do len dau day arr[0...4]
5 **11** 25 12 22 64
6 // Tim phan tu be nhat trong day arr[1...4]
7 // dua phan tu do len dau day arr[1...4]
8 11 **12** 25 22 64
9 // Tim phan tu be nhat trong day arr[2...4]
10 // dua phan tu do len dau day arr[2...4]
11 11 12 **22** 25 64
12 // Tim phan tu be nhat trong day arr[3...4]
13 // dua phan tu do len dau day arr[3...4]
14 11 12 22 **25** 64
```

Từ các bước trên ta thu được dãy được sắp xếp là arr[] = 11 12 22 25 64.

Dựa trên ví dụ đã đưa ra, nhóm chúng em đã thực hiện thuật toán bằng ngôn ngữ C++ như sau :

```
int i, j, min_idx;
// Di chuyển ranh giới của mảng đã sắp xếp và chưa sắp xếp
for (i = 0; i < n - 1; i++) {
    // Tìm phần tử nhỏ nhất trong mảng chưa sắp xếp
    min_idx = i;
    for (j = i + 1; j < n; j++) {
        if (arr[j] < arr[min_idx])
            min_idx = j;
    }
    // Đổi chỗ phần tử nhỏ nhất với phần tử đầu tiên của mảng chưa sắp xếp
    if (min_idx != i) { // Chỉ in ra các bước có sự thay đổi vị trí
        swap(arr[min_idx], arr[i]);
        print(arr);
    }
}
```

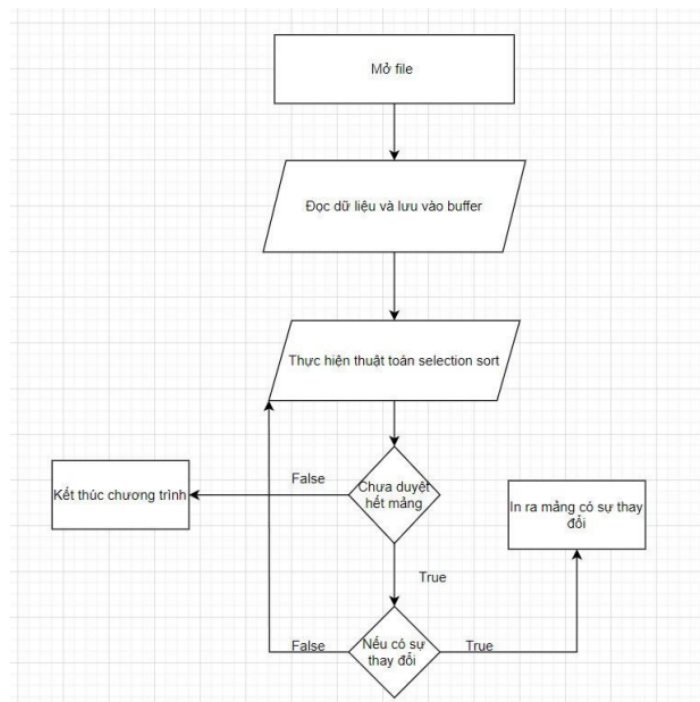
Hình 2: Hiện thực thuật toán Selection Sort bằng C++

Đánh giá thuật toán: Đây là giải thuật đơn giản, dễ hiểu và có thể hiện thực trên nhiều ngôn ngữ lập trình khác nhau; đi kèm với đó là độ hiệu quả không cao hay chính xác hơn là độ phức tạp khá lớn $O(n^2)$.

2. Giải pháp hiện thực :

Sử dụng mã MIPS để hiện thực thuật toán Selection sort.

a. Phân tích luồng thực thi



Hình 3: Lưu đồ thuật toán Selection Sort

Khi chạy chương trình file INT.BIN (input file) sẽ được mở và đọc 40 bytes (tương ứng với 10 số nguyên 4 bytes) và lưu vào 1 buffer (đặt tên là array). Sau đó tiến hành chạy giải thuật selection sort, sau mỗi bước lặp sẽ kiểm tra nếu có thay đổi (SWAP) thì gọi hàm PRINT_ARRAY. Sau khi chạy xong giải thuật thì sẽ đóng file và gọi END_MAIN kết thúc chương trình.



b. Các hàm sử dụng để giải quyết bài toán :

Hàm PRINT_ARRAY :

- Chức năng : dùng để in array hiện tại.
- Đầu vào : array, n.
- Sử dụng các thanh ghi a0, a1, v0, t0, t1 và label của các chuỗi cần in.
v0 : lưu syscall code của lệnh syscall cho print string, integer.
a0 : lưu thông tin cần in trong lệnh syscall.
a1 : lưu địa chỉ của mảng (array).
t1 : chỉ số hiện tại của bước lặp (i).
t0 : lưu kích thước của mảng (n).
- Đầu ra : mảng được in ra Console.

Hàm SWAP :

- Chức năng : hoán đổi giá trị của arr[a1] và arr[a2].
- Đầu vào : array, thanh ghi a1 và a2 là vị trí các phần tử trong mảng.
- Sử dụng các thanh ghi a0, t0, t1, t2, t3, t4. a0 : lưu địa chỉ của mảng (array).
t1 : địa chỉ của phần tử arr[a1].
t2 : địa chỉ của phần tử arr[a2].
t3 : giá trị của phần tử arr[a1].
t4 : giá trị của phần tử arr[a2].
- Đầu ra : không có.



c. Thống kê số lệnh, kiểu lệnh :

Số thứ tự	Lệnh	Kiểu lệnh	Chú thích
1	add	R - type	Add
2	sub	R - type	Sub
3	slt	R - type	Set Less Than
4	jr	R - type	Jump Register
5	syscall	R - type	Syscall
6	bgez	R - type	Branch if Greater Than or Equal to Zero
7	slt	R - type	Set on Less Than
8	mul	R - type	Multiply
9	j	J - type	Jump
10	jal	J - type	Jump and Link
11	beq	I - type	Branch On Equal
12	bge	I - type	Branch if Greater Than or Equal
13	addi	I - type	Add Immediate
14	bne	I - type	Branch On Not Equal
15	lw	I - type	Load Word
16	sw	I - type	Store Word
17	lb	I - type	Load Byte
18	sb	I - type	Store Byte
19	la	Pseudon Instruction	Load Address
20	beqz	Pseudon Instruction	Branch If Equal Zero
21	move	Pseudon Instruction	Move
22	li	Pseudon Instruction	Load Immediate

Bảng 1: Bảng thống kê số lệnh, loại lệnh được sử dụng

III. Chạy thử chương trình các testcase:

a. Một số testcase sử dụng :

STT	Testcase	Phân loại
1	-92 -63 -63 -32 -29 15 26 57 57 99	Best case
2	60 31 22 12 11 -7 -13 -19 -24 -43	Worst case
3	51 6 14 38 -31 14 79 -29 -50 -74	Normal case

Bảng 2: Các testcase mẫu

b. Kết quả và thời gian thực thi của chương trình :

Ta có công thức tính thời gian thực thi như sau :

$$ExcutionTime = \frac{InstructionCount \times CyclesperInstruction}{ClockRate}$$

Cấu hình của máy: CPI = 1 cho 1 lệnh MIPS chuẩn, chương trình trên máy tính kiến trúc MIPS có tần số 1 GHz.

Testcase 1 :

Input : -92 -63 -63 -32 -29 15 26 57 57 99

Output :

Trường hợp tốt nhất khi mảng đã sắp xếp tăng dần, nên sẽ không có sự thay đổi do đó sẽ không in ra console.

	R - type	I - type	J - type
Số lệnh	293	590	74

Thời gian thực thi :

$$ExcutionTime = \frac{InstructionCount \times CyclesperInstruction}{ClockRate} = \frac{957}{1.10^9} = 0,957(\mu s)$$

Testcase 2 :

Input : 60 31 22 12 11 -7 -13 -19 -24 -43

Output :



Mảng ban đầu : 60 31 22 12 11 -7 -13 -19 -24 -43

Các bước có sự thay đổi :

-43 31 22 12 11 -7 -13 -19 -24 60

-43 -24 22 12 11 -7 -13 -19 31 60

-43 -24 -19 12 11 -7 -13 22 31 60

-43 -24 -19 -13 11 -7 12 22 31 60

-43 -24 -19 -13 -7 11 12 22 31 60

Mảng đã được sắp xếp : -43 -24 -19 -13 -7 11 12 22 31 60

	R - type	I - type	J - type
Số lệnh	478	1335	134

Thời gian thực thi :

$$ExcutionTime = \frac{InstructionCount \times CyclesperInstruction}{ClockRate} = \frac{1947}{1.10^9} = 1,947(\mu s)$$

Testcase 3 :

Input : 51 6 14 38 -31 14 79 -29 -50 -74

Output :

Mảng ban đầu : 51 6 14 38 -31 14 79 -29 -50 -74

Các bước có sự thay đổi :

-74 6 14 38 -31 14 79 -29 -50 51

-74 -50 14 38 -31 14 79 -29 6 51

-74 -50 -31 38 14 14 79 -29 6 51

-74 -50 -31 -29 14 14 79 38 6 51

-74 -50 -31 -29 6 14 79 38 14 51

-74 -50 -31 -29 6 14 14 38 79 51

-74 -50 -31 -29 6 14 14 38 51 79

Mảng đã được sắp xếp : -74 -50 -31 -29 6 14 14 38 51 79

	R - type	I - type	J - type
Số lệnh	530	1633	158

Thời gian thực thi :

$$ExcutionTime = \frac{InstructionCount \times CyclesperInstruction}{ClockRate} = \frac{2321}{1.10^9} = 2,321(\mu s)$$



IV. Cài đặt và chạy code :

Nhóm sử dụng phần mềm Mars Mips ver 4.5 chạy trên nền java 20.0.1.

Đầu tiên chạy file taofile.asm để tạo file INT10.BIN, sau đó sẽ chạy file Mn_059.asm để thực hiện thuật toán Selection Sort số nguyên.

Qua bài tập lớn này giúp chúng em cải thiện nhiều hơn về môn kiến trúc máy tính, nhóm cũng cảm ơn thầy Nguyễn Xuân Minh đã hỗ trợ nhóm trong quá trình hoàn thành bài tập lớn. Nếu có sự sai sót trong quá trình làm việc, mong thầy bỏ qua và nhóm sẽ cải thiện trong những bài tập lớn sau.