

ĐỒ ÁN 2

MIPS Architecture and Assembly Language

Quy định chung

- Đồ án được làm theo nhóm: *mỗi nhóm 2-3 sinh viên.*
- Bài làm giống nhau giữa các nhóm, tất cả các nhóm liên quan đều bị điểm **0 cả thực hành**
- Ngôn ngữ sử dụng: **MIPS**

Yêu cầu

Bài 1. (5đ)

Viết chương trình hợp ngữ MIPS 32 bits cho phép người dùng nhập vào vào mảng số nguyên có n ($0 < n < 1000$, nếu không thỏa yêu cầu người dùng nhập lại) phần tử. Hiển thị Menu như bên dưới để người dùng chọn chức năng, thực hiện xong chức năng thì hiển thị lại Menu để người dùng chọn tiếp. Lưu ý: các chức năng phải được viết theo dạng hàm con.

1. Xuất ra các phần tử.
2. Tính tổng các phần tử.
3. Liệt kê các phần tử là số nguyên tố.
4. Tìm max.
5. Tìm phần tử có giá trị x (người dùng nhập vào) trong mảng.
6. Thoát chương trình

Bài 2. (5đ)

Thuật toán Quick sort:

Giải thuật sắp xếp nhanh (Quick Sort) là một giải thuật hiệu quả cao và dựa trên việc chia mảng dữ liệu thành các mảng nhỏ hơn. Giải thuật sắp xếp nhanh chia mảng thành hai phần bằng cách so sánh từng phần tử của mảng với một phần tử được chọn gọi là **phần tử chốt (Pivot)**: một mảng bao gồm các phần tử nhỏ hơn hoặc bằng phần tử chốt và mảng còn lại bao gồm các phần tử lớn hơn hoặc bằng phần tử chốt.

Tiến trình chia này diễn ra tiếp tục cho tới khi độ dài của các mảng con đều bằng 1. Giải thuật sắp xếp nhanh tỏ ra khá hiệu quả với các tập dữ liệu lớn khi mà độ phức tạp trường hợp trung bình và trường hợp xấu nhất là $O(n \log n)$ với n là số phần tử.

Nguồn tham khảo:

<https://en.wikipedia.org/wiki/Quicksort>

<https://vietjack.com/cau-truc-du-lieu-va-giai-thuat/giai-thuat-sap-xep-nhanh.jsp>

<https://www.stdio.vn/articles/quick-sort-546>

<https://viblo.asia/p/gioi-thieu-ve-thuat-toan-quick-sort-Az45bbj65xY>

Giải thuật:

Bước 1: Chọn phần tử chốt là phần tử có chỉ mục cao nhất (phần tử ở cuối danh sách)
Bước 2: Khai báo hai biến để trỏ tới bên trái và bên phải của danh sách, ngoại trừ phần tử chốt
Bước 3: Biến bên trái trỏ tới mảng con bên trái
Bước 4: Biến bên phải trỏ tới mảng con bên phải
Bước 5: Khi giá trị tại biến bên trái là nhỏ hơn phần tử chốt thì di chuyển sang phải
Bước 6: Khi giá trị tại biến bên phải là lớn hơn phần tử chốt thì di chuyển sang trái
Bước 7: Nếu không trong trường hợp cả bước 5 và bước 6 thì trao đổi giá trị biến trái và phải
Bước 8: Nếu $left \geq right$, thì đây chính là giá trị chốt mới

Giải thuật mẫu

```
Bắt đầu hàm partitionFunc (left, right, pivot)

    leftPointer = left - 1
    rightPointer = right

    while True thực hiện
        while A[++leftPointer] < pivot thực hiện
            //không làm điều gì
        kết thúc while

        while rightPointer > 0 && A[--rightPointer] > pivot thực hiện
            //không làm điều gì
        kết thúc while

        if leftPointer >= rightPointer
            break
        else
            Trao đổi leftPointer, rightPointer
    kết thúc if
```

```
kết thúc while
```

```
Tráo đổi leftPointer, right
```

```
return leftPointer
```

```
Kết thúc hàm
```

Yêu cầu chương trình

1. Gồm có 2 chương trình hợp ngữ MIPS: *Array.asm* và *QuickSort.asm* (ngoài ra, còn các file macro đính kèm (nếu có)).
2. Dữ liệu được đọc từ và xuất ra file,

Dữ liệu input và output thuật giải QuickSort.

- a. Cấu trúc file input_sort.txt:

- i. n: số lượng phần tử (với $2 \leq n \leq \text{max}=1000$ phần tử)
- ii. array_int: dãy số nguyên dương (mỗi số nguyên cách nhau bởi khoảng trắng)
ví dụ:

10

34 21 43 29 21 453 1 12 0 34

- b. Cấu trúc file output_sort.txt

- i. array_int: dãy số nguyên đã được sắp xếp tăng dần

Hình thức nộp và chấm bài:

- File nộp: **MSSV1_MSSV2.zip**
 - Trong đó có 2 file: *Array.asm* và *QuickSort.asm* (ngoài ra, còn các file macro đính kèm (nếu có)).
 - File báo cáo MSSV1_MSSV2_MSSV3.pdf: Trình bày phân công công việc, đánh giá mức độ hoàn thành, các test case