

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO TIẾN ĐỘ TUẦN

Môn học: Project 1

Chủ đề: Ứng dụng thuật toán tìm kiếm

Tuần: 18/11 – 23/11

Giáo viên hướng dẫn:

Lê Bá Vui

Sinh viên thực hiện:

Lê Minh Triết

Hà Nội - 2024

Contents

1. CÔNG VIỆC ĐÃ HOÀN THÀNH TRONG TUẦN	3
1.1. Lựa chọn các thuật toán tìm kiếm	3
1.2. Tạo khung chính cho dự án	3
1.3. Thiết kế ban đầu cho class SearchAlgorithm	3
1.4. Thiết kế ban đầu cho class NeedSortedAlgorithm	4
1.5. Triển khai các thuật toán LinearSearch, BinarySearch	4
1.5.1. LinearSearch	4
1.5.2. BinarySearch	4
1.5.3. Đánh giá kết quả	4
2. DỰ KIẾN CÔNG VIỆC TUẦN SAU	6
3. SOURCE CODE	6

1. Công việc đã hoàn thành trong tuần

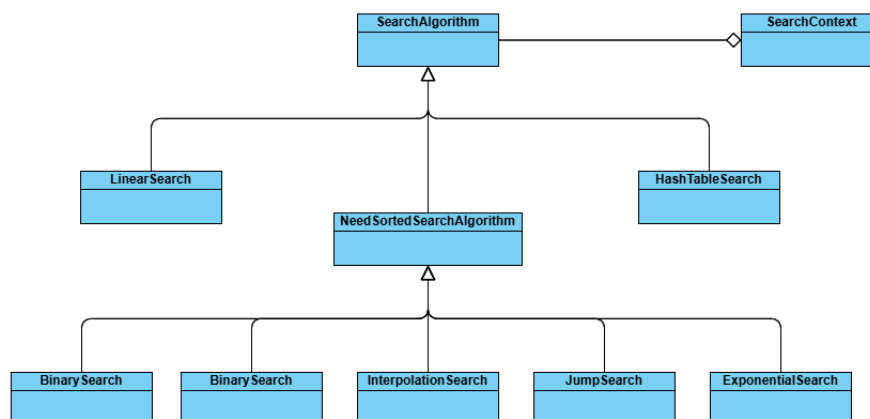
1.1. Lựa chọn các thuật toán tìm kiếm

7 thuật toán tìm kiếm cơ bản được lựa chọn

1. Tìm kiếm tuần tự (Linear Search).
2. Tìm kiếm bằng bảng băm (Hashing).
3. Tìm kiếm nhị phân (Binary Search).
4. Tìm kiếm nhảy (Jump Search).
5. Tìm kiếm Exponential (Exponential Search).
6. Tìm kiếm Ternary (Ternary Search).
7. Tìm kiếm nhị phân theo Interpolation (Interpolation Search).

1.2. Tạo khung chính cho dự án

Dự án đi theo mô hình hướng đối tượng, sẽ gồm các class chính được miêu tả trong biểu đồ sau



Hình 1. Biểu đồ Class cho Project

Trong đó, SearchAlgorithm là lớp cha cho mọi thuật toán tìm kiếm, và SearchContext là class để quyết định lớp thuật toán tìm kiếm nào trong quá trình chạy sẽ được gọi tới (sử dụng Strategy Pattern).

Ở các con của SearchAlgorithm có một lớp con NeedSortedSearchAlgorithm đại diện cho các thuật toán tìm kiếm phải sắp xếp lại dãy.

1.3. Thiết kế ban đầu cho class SearchAlgorithm

Các thuộc tính cơ bản:

- arr: một vector<int> lưu lại danh sách cần tìm kiếm.
- Các thuộc tính để lưu lại thời gian chạy (bao gồm tổng thời gian chạy, thời gian xây dựng các cấu trúc dữ liệu (bảng băm với HashTableSearch, hoặc là thời gian xây dựng mảng đã sắp xếp với các thuật toán cần sắp xếp) và thời gian tìm kiếm.

Các hàm quan trọng

- int search(int value): tìm kiếm một giá trị và trả về vị trí.

- void printInfo(): In ra thời gian chạy để tìm kiếm (bao gồm các loại thời gian trên).
- void setArr(vector<int> arr): Đưa dãy cần tìm kiếm vào.

1.4. Thiết kế ban đầu cho class NeedSortedAlgorithm

Các thuộc tính cơ bản

- isSorted: đánh dấu xem đã sắp xếp lại dãy cần tìm kiếm chưa.
- sortedIndexArr: một vector chứa các pair lưu giữ mảng sau khi sắp xếp và chỉ số cũ trong mảng ban đầu.

Các phương thức quan trọng

- bool isAscending(vector<T> arr): Kiểm tra dãy có tăng không.
- void buildSortedArr(): Xây dựng sortedIndexArr.

1.5. Triển khai các thuật toán LinearSearch, BinarySearch

1.5.1. LinearSearch

Yêu cầu: Không

Thuật toán: Duyệt qua từng phần tử của dãy, từ đầu tới cuối tới khi tìm thấy hoặc kết thúc dãy.

Độ phức tạp: Tối nhất là $O(n)$.

1.5.2. BinarySearch

Yêu cầu: Dãy đã được sắp xếp

Thuật toán: So sánh với giá trị nằm giữa dãy. Nếu đây là giá trị cần tìm thì trả về. Nếu giá trị cần tìm nhỏ hơn thì thực hiện tương tự với dãy ở nửa đầu. Ngược lại tìm ở nửa sau của dãy.

Độ phức tạp: Trung bình là $O(\log n)$.

1.5.3. Đánh giá kết quả

Đã triển khai thành công hai thuật toán tìm kiếm cơ bản nhất này. Source code đã được đính kèm ở phần dưới.

Đã thử test với 2 dãy sau:

- Test 1: Dãy gồm 1000000 số từ 0 tới 10^6-1 theo thứ tự tăng dần.
- Test 2: Dãy gồm 1000000 số từ 0 tới 10^6-1 theo thứ tự giảm dần.

Với mỗi Test, thực hiện tìm 12 giá trị cách đều nhau là $i \cdot 10^6 / 13$ (lấy phần nguyên, thu được kết quả sau

```

Test 1 -----:
LinearSearch:
Run Time:: Total Time 0.432 ms; Build Time 0.000 ms; Search Time 0.430 ms.Find 76923 at 76923
Run Time:: Total Time 0.807 ms; Build Time 0.000 ms; Search Time 0.807 ms.Find 153846 at 153846
Run Time:: Total Time 1.378 ms; Build Time 0.000 ms; Search Time 1.369 ms.Find 230769 at 230769
Run Time:: Total Time 1.098 ms; Build Time 0.000 ms; Search Time 1.097 ms.Find 307692 at 307692
Run Time:: Total Time 1.990 ms; Build Time 0.000 ms; Search Time 1.990 ms.Find 384615 at 384615
Run Time:: Total Time 1.697 ms; Build Time 0.000 ms; Search Time 1.696 ms.Find 461538 at 461538
Run Time:: Total Time 1.718 ms; Build Time 0.000 ms; Search Time 1.718 ms.Find 538461 at 538461
Run Time:: Total Time 2.368 ms; Build Time 0.000 ms; Search Time 2.367 ms.Find 615384 at 615384
Run Time:: Total Time 3.262 ms; Build Time 0.000 ms; Search Time 3.261 ms.Find 692307 at 692307
Run Time:: Total Time 3.919 ms; Build Time 0.000 ms; Search Time 3.917 ms.Find 769230 at 769230
Run Time:: Total Time 3.106 ms; Build Time 0.000 ms; Search Time 3.105 ms.Find 846153 at 846153
Run Time:: Total Time 2.901 ms; Build Time 0.000 ms; Search Time 2.901 ms.Find 923076 at 923076
BinarySearch:
Run Time:: Total Time 47.324 ms; Build Time 47.319 ms; Search Time 0.002 ms.Find 76923 at 76923
Run Time:: Total Time 0.003 ms; Build Time 0.000 ms; Search Time 0.002 ms.Find 153846 at 153846
Run Time:: Total Time 0.002 ms; Build Time 0.000 ms; Search Time 0.002 ms.Find 230769 at 230769
Run Time:: Total Time 0.003 ms; Build Time 0.000 ms; Search Time 0.002 ms.Find 307692 at 307692
Run Time:: Total Time 0.002 ms; Build Time 0.000 ms; Search Time 0.001 ms.Find 384615 at 384615
Run Time:: Total Time 0.002 ms; Build Time 0.000 ms; Search Time 0.001 ms.Find 461538 at 461538
Run Time:: Total Time 0.002 ms; Build Time 0.000 ms; Search Time 0.001 ms.Find 538461 at 538461
Run Time:: Total Time 0.002 ms; Build Time 0.000 ms; Search Time 0.001 ms.Find 615384 at 615384
Run Time:: Total Time 0.001 ms; Build Time 0.000 ms; Search Time 0.001 ms.Find 692307 at 692307
Run Time:: Total Time 0.002 ms; Build Time 0.000 ms; Search Time 0.002 ms.Find 769230 at 769230
Run Time:: Total Time 0.002 ms; Build Time 0.000 ms; Search Time 0.002 ms.Find 846153 at 846153
Run Time:: Total Time 0.002 ms; Build Time 0.000 ms; Search Time 0.001 ms.Find 923076 at 923076

```

Hình 2. Test 1

```

Test 2 -----:
LinearSearch:
Run Time:: Total Time 4.010 ms; Build Time 0.000 ms; Search Time 4.006 ms.Find 76923 at 923078
Run Time:: Total Time 4.344 ms; Build Time 0.000 ms; Search Time 4.341 ms.Find 153846 at 846155
Run Time:: Total Time 3.062 ms; Build Time 0.000 ms; Search Time 3.060 ms.Find 230769 at 769232
Run Time:: Total Time 3.233 ms; Build Time 0.000 ms; Search Time 3.231 ms.Find 307692 at 692309
Run Time:: Total Time 3.349 ms; Build Time 0.000 ms; Search Time 3.347 ms.Find 384615 at 615386
Run Time:: Total Time 2.622 ms; Build Time 0.000 ms; Search Time 2.621 ms.Find 461538 at 538463
Run Time:: Total Time 1.648 ms; Build Time 0.000 ms; Search Time 1.647 ms.Find 538461 at 461540
Run Time:: Total Time 1.985 ms; Build Time 0.000 ms; Search Time 1.984 ms.Find 615384 at 384617
Run Time:: Total Time 1.616 ms; Build Time 0.000 ms; Search Time 1.614 ms.Find 692307 at 307694
Run Time:: Total Time 1.509 ms; Build Time 0.000 ms; Search Time 1.494 ms.Find 769230 at 230771
Run Time:: Total Time 0.804 ms; Build Time 0.000 ms; Search Time 0.802 ms.Find 846153 at 153848
Run Time:: Total Time 0.494 ms; Build Time 0.000 ms; Search Time 0.492 ms.Find 923076 at 76925
BinarySearch:
Run Time:: Total Time 168.257 ms; Build Time 168.252 ms; Search Time 0.003 ms.Find 76923 at 923078
Run Time:: Total Time 0.002 ms; Build Time 0.000 ms; Search Time 0.001 ms.Find 153846 at 846155
Run Time:: Total Time 0.002 ms; Build Time 0.000 ms; Search Time 0.001 ms.Find 230769 at 769232
Run Time:: Total Time 0.003 ms; Build Time 0.000 ms; Search Time 0.002 ms.Find 307692 at 692309
Run Time:: Total Time 0.002 ms; Build Time 0.000 ms; Search Time 0.001 ms.Find 384615 at 615386
Run Time:: Total Time 0.057 ms; Build Time 0.000 ms; Search Time 0.001 ms.Find 461538 at 538463
Run Time:: Total Time 0.003 ms; Build Time 0.000 ms; Search Time 0.002 ms.Find 538461 at 461540
Run Time:: Total Time 0.002 ms; Build Time 0.000 ms; Search Time 0.001 ms.Find 615384 at 384617
Run Time:: Total Time 0.003 ms; Build Time 0.000 ms; Search Time 0.002 ms.Find 692307 at 307694
Run Time:: Total Time 0.002 ms; Build Time 0.000 ms; Search Time 0.001 ms.Find 769230 at 230771
Run Time:: Total Time 0.002 ms; Build Time 0.000 ms; Search Time 0.001 ms.Find 846153 at 153848
Run Time:: Total Time 0.003 ms; Build Time 0.000 ms; Search Time 0.001 ms.Find 923076 at 76925

```

Hình 3. Test 2

Nhận xét ban đầu:

Table 1

Test	Search Algorithm	Thời gian tốt nhất (ms)		Thời gian trung bình (ms)		Thời gian tồi nhất (ms)	
		Total	Search	Total	Search	Total	Search
1	Linear	0.432	0.430	2.056	2.055	3.919	3.917
1	Binary	0.001	0.001	3.946	0.002	47.319	0.002
2	Linear	0.494	0.492	2.389	2.387	4.344	4.341
2	Binary	0.002	0.001	14.028	0.001	168.257	0.003

Do vậy, có thể nhận xét rằng:

- Binary Search mất thời gian chủ yếu khi build (khi sắp xếp lại dãy). Còn lại thời gian tìm kiếm luôn nhỏ hơn 2000 lần so với Linear Search, điều này là phù hợp do độ phức tạp của Binary Search là $O(\log n)$ còn của LinearSearch là $O(n)$.
- Nếu chỉ có 1 truy vấn tìm kiếm cho dãy, Linear Search hoạt động hiệu quả hơn, nhưng nếu có nhiều truy vấn thì BinarySearch là hiệu quả hơn rất nhiều.

2. Dự kiến công việc tuần sau

- Thiết lập giao diện để tương tác với người dùng trên Terminal.
- Triển khai các thuật toán Ternary Search và Jump Search.

3. Source code

Source code nằm ở link github bên dưới:

<https://github.com/trietp1253201581/SearchApp>