

Cấu trúc dữ liệu và thuật toán ảnh hưởng thế nào đến hiệu năng phần mềm?

How Data Structures and Algorithms Affect Real-World Software Performance



1 Chen Baode

2 Hoàng Trọng Tấn

3 Nguyễn Sơn

4 Phan Hồng Trọng

5 Nguyễn Quang Minh

Mục lục



1



Thuật toán và cấu trúc dữ liệu

(Algorithms and Data Structures)



Chen Baode

2



Thuật toán tìm kiếm

(Searching Algorithms)



Hoàng Trọng Tấn

3



Thuật toán sắp xếp

(Sorting Algorithms)



Nguyễn Sơn

4



Cấu trúc dữ liệu tuyến tính

(Array, Linked List, Stack, Queue)



Phan Hồng Trọng

5



Cây, đồ thị, băm và kết luận

(Tree, Graph, Hashing, Conclusion)



Nguyễn Quang Minh

01

Thuật toán và cấu trúc dữ liệu

(Algorithms and Data Structures)



Người trình bày

Chen Baode



Phần mềm nhanh hay chậm do đâu?



1



Cấu trúc dữ liệu

(Data Structure)

Dữ liệu được tổ chức thế nào



Thuật toán

(Algorithm)

Dữ liệu được xử lý thế nào



Hiệu năng (Performance)

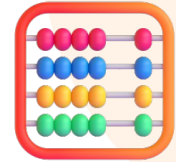


Thời gian chạy



Bộ nhớ sử dụng

Thuật toán (Algorithm)



1

Đầu vào (Input)

Xử lý từng bước

Đầu ra (Output)

Ví dụ: tìm số lớn nhất (Find Largest)

Danh sách

12

8

13

9

11

Largest

12

12

13

13

13

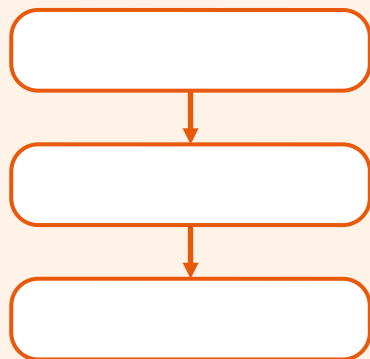


Lặp: nếu số hiện tại > Largest thì gán Largest = số hiện tại

Ba cấu trúc cơ bản của thuật toán



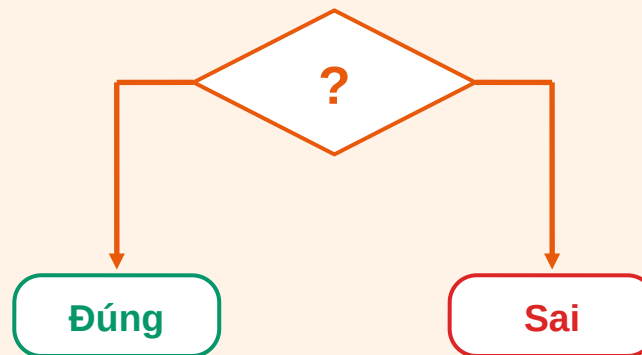
1



Tuần tự

(Sequence)

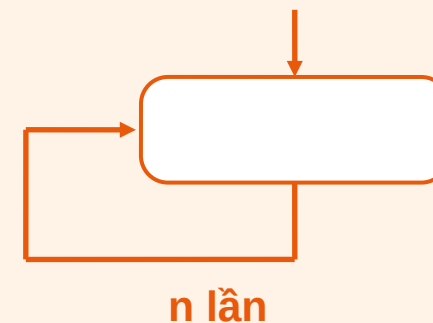
Các lệnh chạy lần lượt



Quyết định

(Decision)

Kiểm tra điều kiện, chọn nhánh



Lặp

(Repetition)

Lặp lại một nhóm lệnh



Vòng lặp chạy n lần $\rightarrow$ thời gian tăng theo n

Cấu trúc dữ liệu (Data Structure)



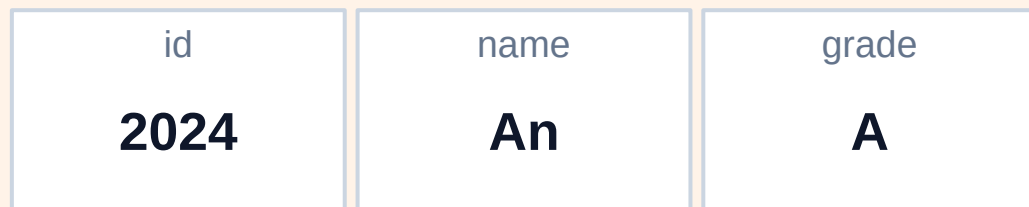
1

Cách tổ chức dữ liệu để lưu trữ và xử lý



Mảng (Array)

Các phần tử cùng kiểu, truy cập bằng chỉ số (index)



Bản ghi (Record)

Nhiều trường (field) khác kiểu, chung một tên



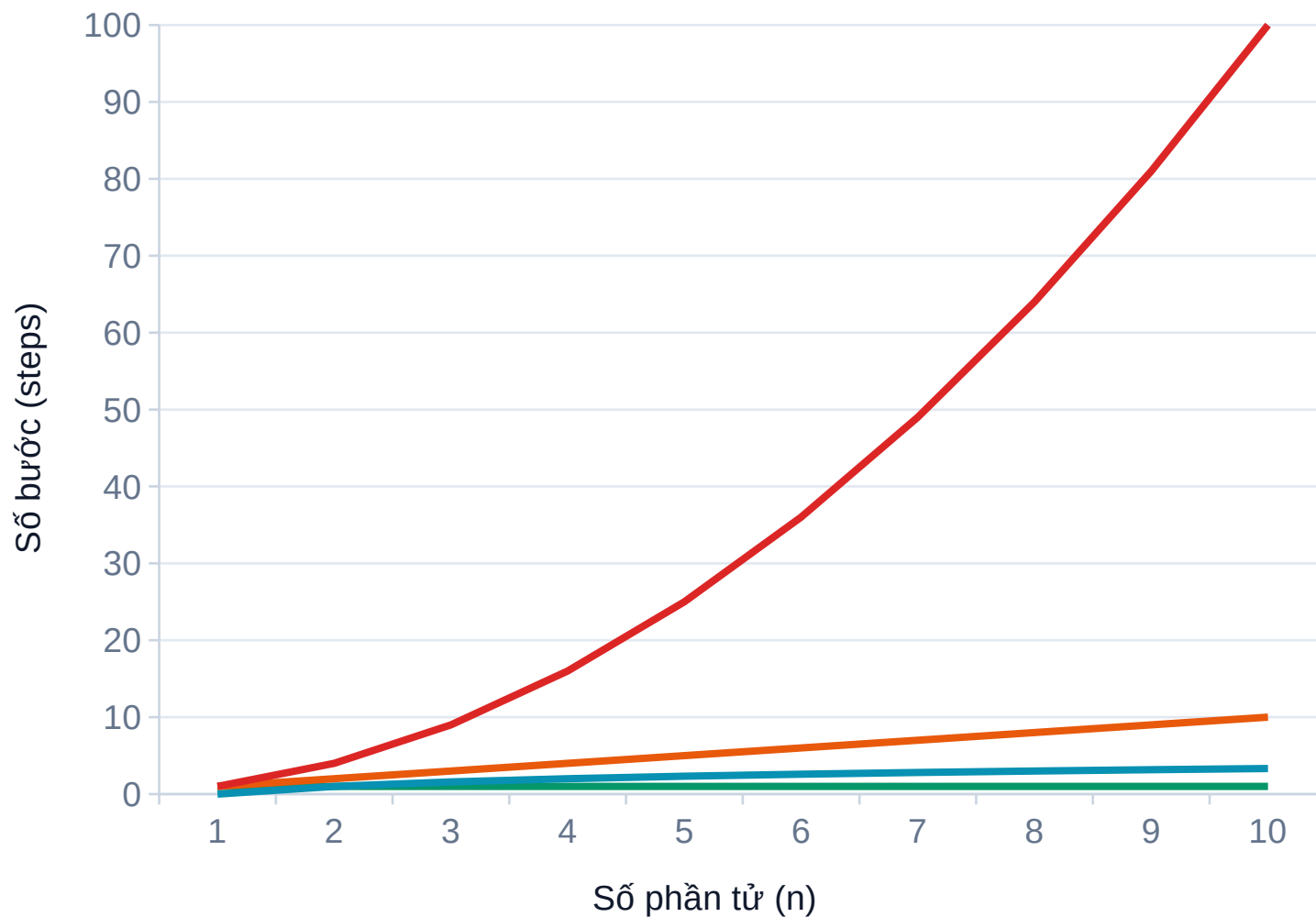
Danh sách liên kết (Linked List)

Mỗi nút (node) gồm dữ liệu và liên kết (link)

Đo hiệu năng: số bước khi n tăng



1



$O(n^2)$

Bình phương (quadratic)

Sắp xếp chọn, nổi bọt



$O(n)$

Tuyến tính (linear)

Tìm kiếm tuần tự



$O(\log n)$

Logarit (logarithmic)

Tìm kiếm nhị phân



$O(1)$

Hằng số (constant)

Truy cập mảng theo chỉ số



02

Thuật toán tìm kiếm

(Searching Algorithms)



Người trình bày

Hoàng Trọng Tấn



Tìm kiếm (Searching)

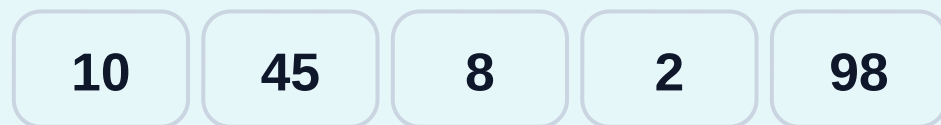


2

Tìm vị trí của giá trị cần tìm (target) trong danh sách

Tìm kiếm tuần tự

(Sequential / Linear Search)



Xét lần lượt từng phần tử



Dùng cho mọi danh sách

Tìm kiếm nhị phân

(Binary Search)



Bắt đầu từ giữa (mid)



Danh sách phải được sắp xếp
(sorted)

Tìm kiếm tuần tự (Linear Search)



2

Tìm 98

10

45

8

2

98

33

11

1

2

3

4

5

6

7

×

×

×

×

✓

5

lần so sánh trong ví dụ này

n

lần so sánh tối đa (worst case)

$(n + 1) / 2$

lần so sánh trung bình (average)

Tìm kiếm nhị phân (Binary Search)



2

Tìm 14

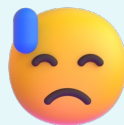
Bước 1 giữa (mid) = 8 · $14 > 8 \rightarrow$ bỏ nửa trái



Bước 2 giữa (mid) = 14 · tìm thấy



Tuần tự (Linear): 9 lần



Nhị phân (Binary): 2 lần



Mỗi bước bỏ một nửa $\rightarrow$ tối đa $\approx \log_2 n$ lần so sánh · **$O(\log n)$**

Số lần so sánh tối đa khi n tăng



2

Số phần tử (n)



Tuần tự (Linear)



Nhị phân (Binary)

1.000

1.000

10

1.000.000

1.000.000

20

1.000.000.000

1.000.000.000

30

Ứng dụng thực tế (Real-world)



Danh bạ điện thoại



Từ điển



Cơ sở dữ liệu (Database)

03

Thuật toán sắp xếp

(Sorting Algorithms)



Người trình bày

Nguyễn Sơn



Vì sao cần sắp xếp (Sorting)?



Chưa sắp xếp

Minh
An
Tấn
Bảo
Sơn
Hà

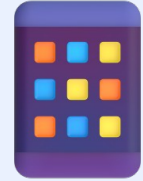


Phải đọc từng dòng



Đã sắp xếp (A → Z)

An
Bảo
Hà
Minh
Sơn
Tấn



Dùng được tìm kiếm nhị phân



Sắp xếp (Sorting): sắp dữ liệu theo thứ tự giá trị

Sắp xếp chọn (Selection Sort)

1 2
3 4

3

1

Tìm phần tử nhỏ nhất ở phần chưa xếp

2

Đổi chỗ với phần tử đầu phần chưa xếp

3

Dời bức tường (wall) sang phải 1 ô

Ban đầu

23

78

45

8

32

56

Lượt 1

8

78

45

23

32

56

Lượt 2

8

23

45

78

32

56

Lượt 3

8

23

32

78

45

56

Lượt 4

8

23

32

45

78

56

Lượt 5

8

23

32

45

56

78

Sắp xếp nổi bọt (Bubble Sort)



3



So sánh 2 phần tử kế nhau từ cuối lên, đổi chỗ nếu sai thứ tự → số nhỏ nhất nổi lên đầu

	 8 nổi lên (bubble up)					
Ban đầu	23	78	45	8	32	56
Lượt 1	8	23	78	45	32	56
Lượt 2	8	23	32	78	45	56
Lượt 3	8	23	32	45	78	56
Lượt 4	8	23	32	45	56	78

Sắp xếp 1.000.000 phần tử



3

Chọn · Nổi bọt · Chèn

(Selection · Bubble · Insertion)

$O(n^2)$

≈ 8 phút



Sắp xếp nhanh · Trộn

(Quick Sort · Merge Sort)

$O(n \log n)$

$\approx 0,02$ giây



Thời gian chạy

$O(n^2)$



$O(n \log n)$

gần như bằng 0



Giả sử máy thực hiện 10^9 phép so sánh mỗi giây

04

Cấu trúc dữ liệu tuyến tính

(Array, Linked List, Stack, Queue)



Người trình bày

Phan Hồng Trọng



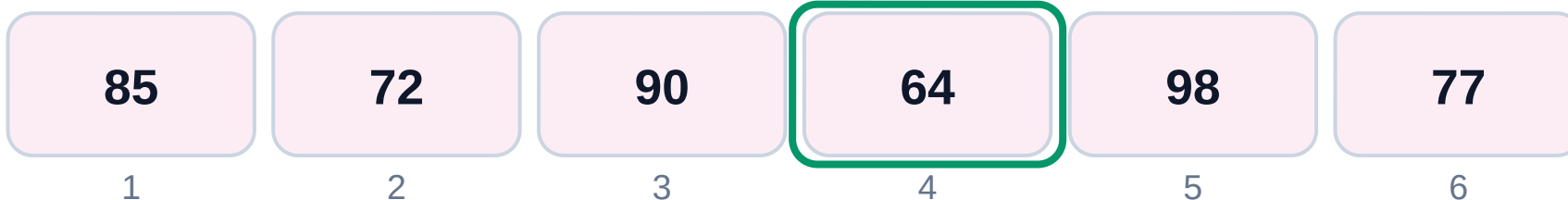
Mảng (Array)



4

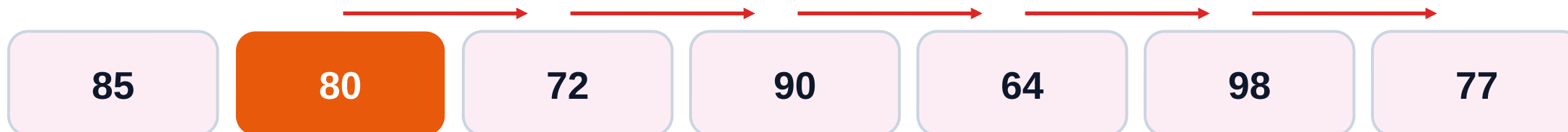
Các phần tử nằm liền kề nhau trong bộ nhớ

scores



`scores[4]` → lấy ngay, 1 bước · $O(1)$

Chèn 80 vào vị trí 2

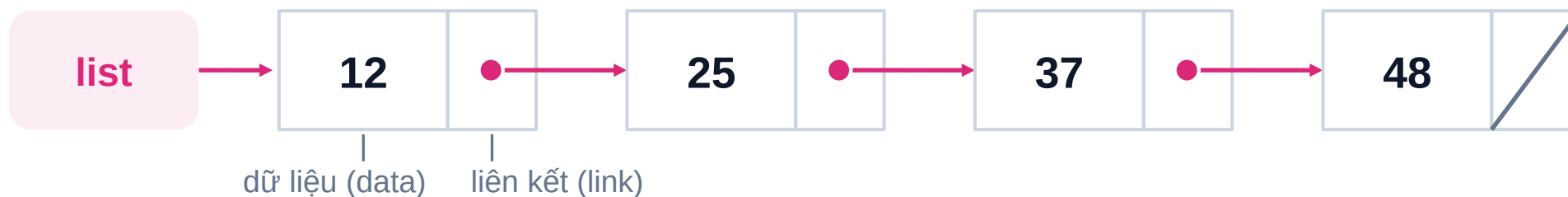


Chèn/xóa ở giữa: phải dời nhiều phần tử · $O(n)$

Danh sách liên kết (Linked List)



4



Chèn 30 sau 25



Chèn/xóa khi đã biết vị trí: chỉ đổi 2 liên kết · $O(1)$



Tìm kiếm: chỉ duyệt tuần tự từ đầu · $O(n)$

Mảng hay danh sách liên kết?



4

Thao tác

Mảng (Array)

Danh sách liên kết (Linked List)

Truy cập phần tử thứ i



Tức thì · $O(1)$



Duyệt từ đầu · $O(n)$

Chèn/xóa ở giữa



Dời phần tử · $O(n)$



Đổi liên kết · $O(1)$

Tìm kiếm nhị phân



Được (khi đã sắp xếp)



Không

Bộ nhớ đệm (cache)



Liên kề → đọc nhanh



Rải rác → chậm hơn

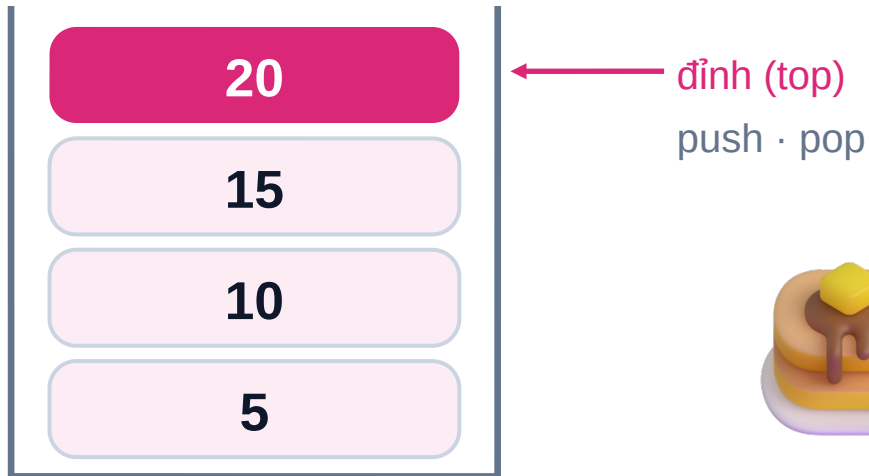


Chọn theo thao tác được dùng nhiều nhất

Ngăn xếp (Stack) và Hàng đợi (Queue)

Ngăn xếp (Stack)

Vào sau – ra trước (LIFO)



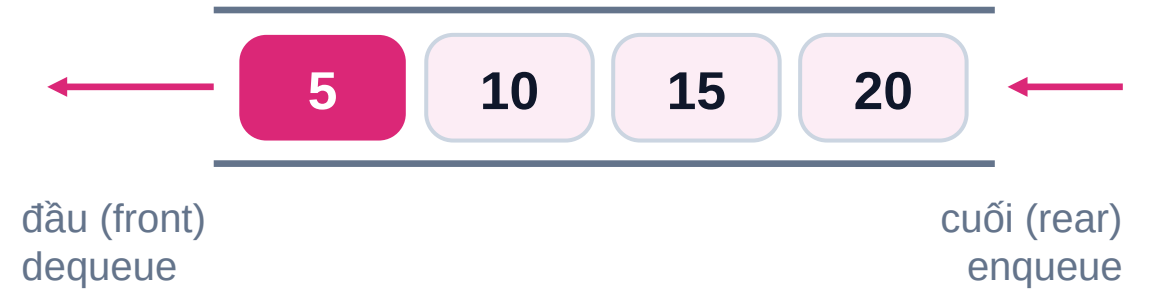
Hoàn tác (Undo)



Nút quay lại (Back)

Hàng đợi (Queue)

Vào trước – ra trước (FIFO)



Hàng đợi in (Print queue)



Lập lịch tiến trình (OS)

push, pop, enqueue, dequeue: mỗi thao tác 1 bước · $O(1)$

05

Cây, đồ thị, băm và kết luận

(Tree, Graph, Hashing, Conclusion)



Người trình bày

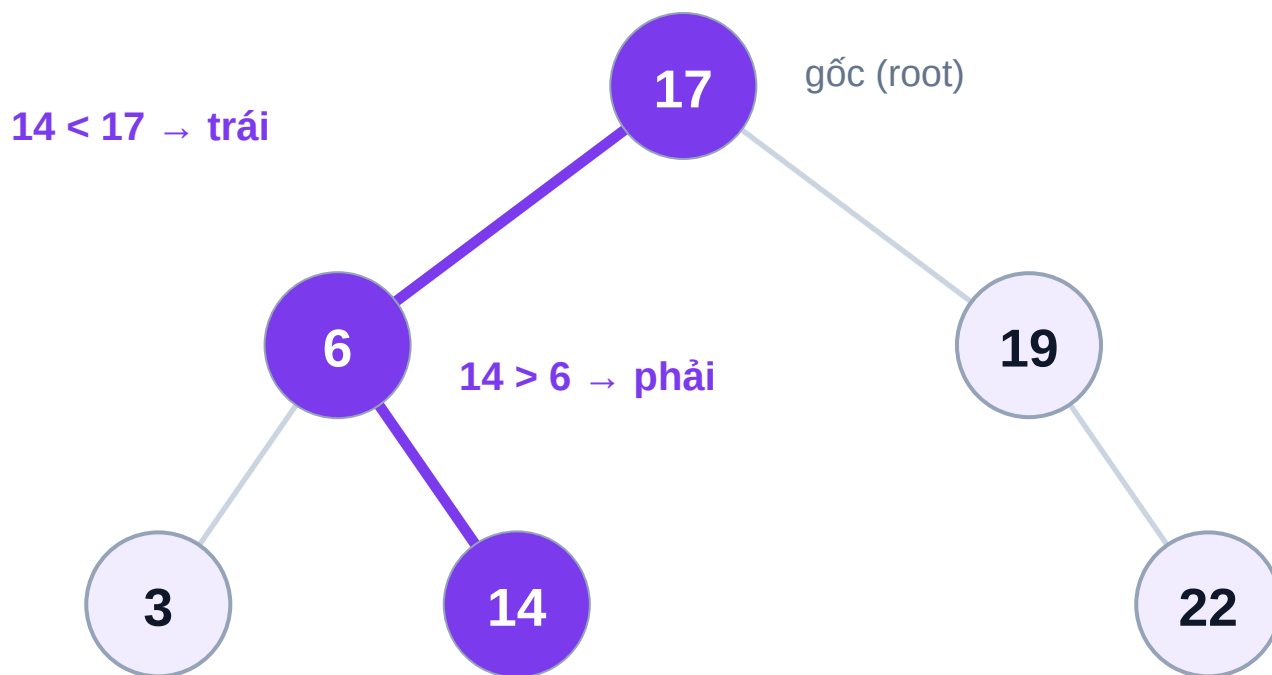
Nguyễn Quang Minh



Cây nhị phân tìm kiếm (BST)



5



Trái < Gốc < Phải (Left < Root < Right)

Tìm 14: 3 bước



Tìm như nhị phân: $\approx \log_2 n$ bước
(cây cân đối)



Chèn/xóa không phải dời phần tử

Ứng dụng



Thư mục (Directory)

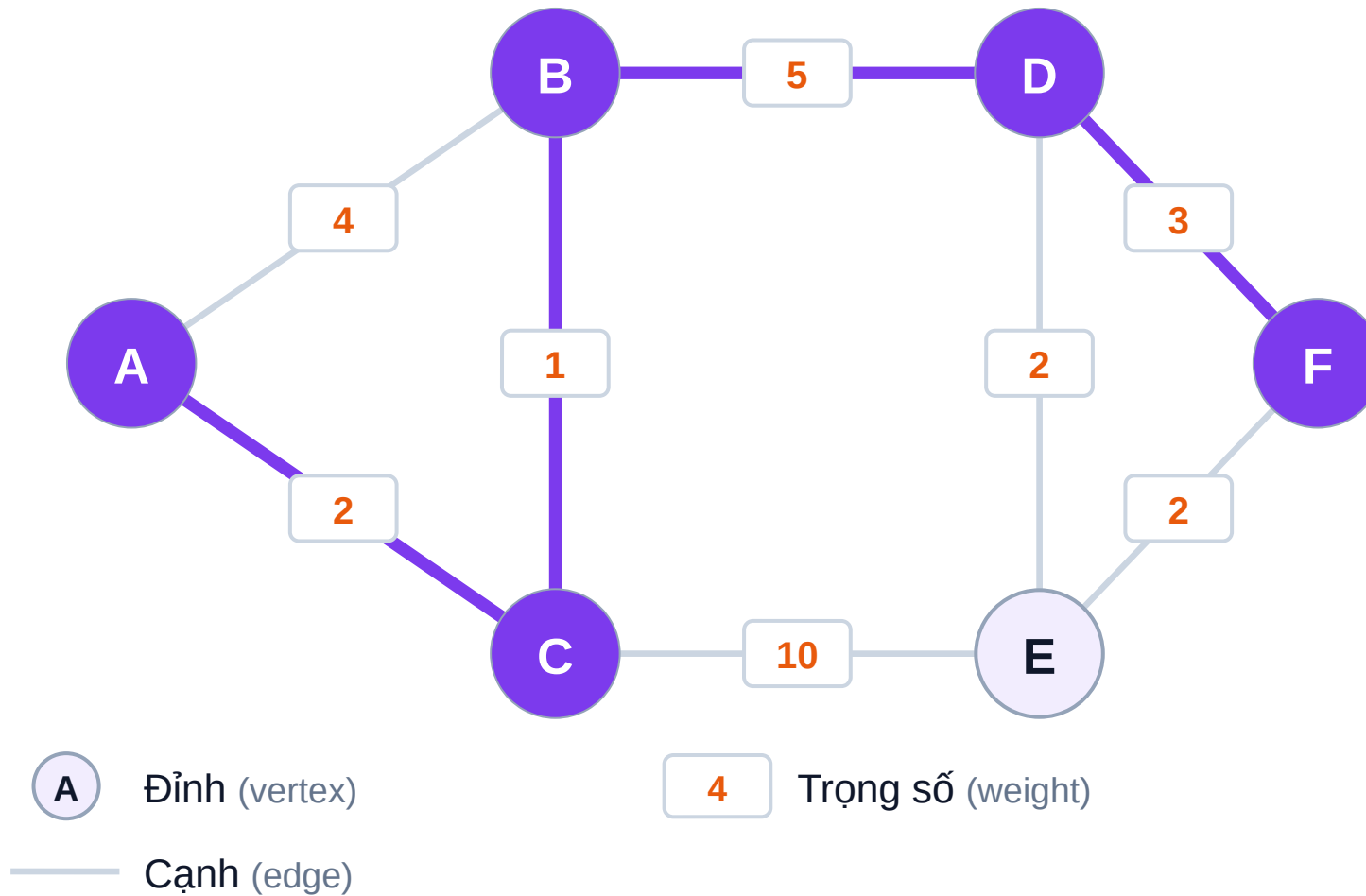


Chỉ mục CSDL (Index)

Đồ thị (Graph)



5



Đường ngắn nhất

A → C → B → D → F
= 2 + 1 + 5 + 3 = 11

Ứng dụng



Bản đồ, GPS tìm đường



Mạng xã hội: gợi ý bạn bè



Mạng máy tính: định tuyến (routing)

Băm (Hashing)



5

Tập tuần tự

(Sequential file)

Đọc lần lượt từ đầu

Tập chỉ mục

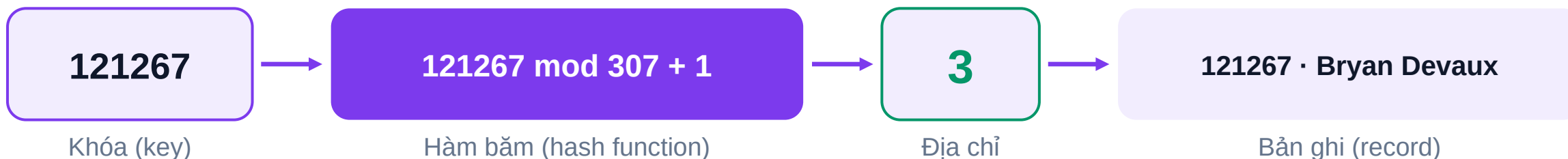
(Indexed file)

Tra chỉ mục → địa chỉ

Tập băm

(Hashed file)

Tính ra địa chỉ ngay



Va chạm (collision): 2 khóa cùng địa chỉ → nối bằng danh sách liên kết



Ứng dụng: tra tài khoản ngân hàng, đăng nhập, bộ nhớ đệm (cache)

Tổng kết



5

Bài toán



Lựa chọn chậm



Lựa chọn nhanh

Tìm 1 phần tử
trong 1.000.000

Tuần tự:
1.000.000 lần

Nhị phân, BST: ≈ 20 lần
Băm: ≈ 1 lần

Sắp xếp
1.000.000 phần tử

Chọn, nổi bọt:
 ≈ 8 phút

Quick sort, merge sort:
 $\approx 0,02$ giây

Chèn liên tục
vào giữa

Mảng:
dời n phần tử

Danh sách liên kết:
đổi 2 liên kết

Lấy phần tử thứ i

Danh sách liên kết:
đi i bước

Mảng:
1 bước

Kết luận



5



Dữ liệu càng lớn, chênh lệch hiệu năng càng lớn



Không có cấu trúc dữ liệu tốt nhất cho mọi việc



Lập trình (Programming) tốt = chọn đúng cấu trúc dữ liệu + thuật toán



Cảm ơn đã lắng nghe

Hỏi & Đáp (Q&A)

1 Chen Baode

2 Hoàng Trọng Tấn

3 Nguyễn Sơn

4 Phan Hồng Trọng

5 Nguyễn Quang Minh