

DEFINITION

Brute Force là gì ?



USAGE

Khi nào cần dùng Brute Force ?

FRAMEWORK

Dạng thuật toán phổ quát



PROS & CONS

Ưu và nhược điểm của Brute Force

O J DEFINITION

Brute Force là gì?



Brute Force là gì?

Brute Force là cách tiếp cận đơn giản để giải quyết bài toán thường trực tiếp dựa trên đề bài và định nghĩa của các khái niệm liên quan.

Brute Force vét cạn và chạy qua tất cả các trường hợp có thể để giải quyết bài toán, cả trường hợp đúng và trường hợp sai.



Độ phức tạp

Phụ thuộc vào giải thuật phương pháp này



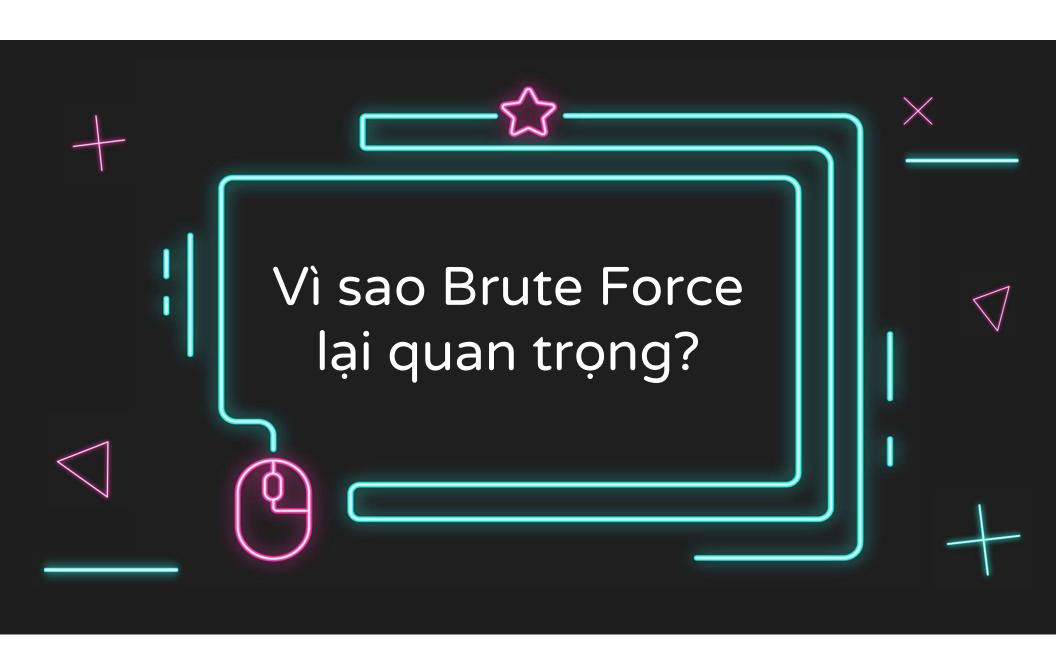
Bản chiếu 6

1

Need to update Thập Huỳnh Nhân, 4/6/2023

Độ phức tạp nếu áp dụng Brute Force vào bài toán:





LÝ DO BRUTE FORCE LẠI QUAN TRỌNG

Đầu tiên

Được sử dụng cho mục đích nghiên cứu và giáo dục (đánh giá hiệu năng của các giải thuật khác)





Thứ hai

Đối với các bài toán như sắp xếp, tìm kiếm và nhân ma trận, Brute Force giúp đưa ra thuật toán hợp lý có một số giá trị thực tiễn, mà không bị giới hạn bởi kích thước đối tượng

LÝ DO BRUTE FORCE LẠI QUAN TRỌNG

Thứ 3

Không cần thiết phải thiết kế ra thuật toán có hiệu năng cao hơn, nếu bài toán có độ phức tạp không quá lớn.

Brute Force vẫn có thể giải quyết bài toán trong khoảng thời gian phù hợp.



Thứ 4

Dù có hiệu năng không cao, Brute Force vẫn có hiệu quả với những bài toán có độ phức tạp nhỏ.



Cuối cùng

Được áp dụng rộng rãi để giải nhiều bài toán.



FRAMEWORK

Dạng thuật toán phổ quát

PSEUDO-CODE

```
c ← first(P)
while c not A do
    if valid(P, c)
    then output(P, c)
    end if
    c ← next(P, c)
end while
```

P là bài toán c là giải pháp khả thi cho P valid(P, c) kiểm tra c là lời giải cho P? output(P, c) dùng c cho P tùy theo ứng dụng first(P) trả về giải pháp c đầu tiên của bài toán P next(P, c) trả về giải pháp c tiếp theo được xét đến của bài toán P, ngay sau giải pháp c được cung cấp trong đối số

USAGE

Khi nào cần dùng Brute Force ?

KHI NÀO CẦN DÙNG?

Bài toán có thể áp dụng Brute Force nếu có đặc điểm sau:

Ví dụ:

Kích thước nhỏ

Sắp xếp mảng không quá 100 phần tử Cần tìm tất cả các giải pháp khả thi để giải quyết bài toán

Tìm giá trị lớn nhất trong một mảng không được sắp xếp

Tuy nhiên không nên dùng Brute Force cho các bài toán



Có thể thấy các bài toán lớn sẽ cần một khoảng thời gian khủng khiếp để giải quyết nếu sử dụng Brute Force Cần tìm ra lời giải mà không phải cần phải duyệt qua tất cả các giải pháp

Tìm một phần tử trong mảng đã sắp xếp theo giá trị cho trước, ta có thể áp dụng giải thuật tìm kiếm nhị phân trong thời gian O(log n)

OA PROS & CONS

Ưu điểm và nhược điểm

Pros:

Có thể áp dụng không giới hạn trong 1 chủ đề nào

Phương pháp lý tưởng cho các bài toán nhỏ và đơn giản

Có thể so sánh với các thuật toán hiệu quả hơn, nhằm nhấn mạnh tính ưu việt của thuật toán đó

> Tìm được tất cả các giải pháp khả thi để giải quyết bài toán

Cons:

Không hiệu quả

Chậm

Phụ thuộc vào hiệu suất của máy tính hơn là thiết kế thuật toán

> Không sáng tạo so với nhiều thuật toán khác



Bài toán thích hợp

Kiểm tra tính nguyên tố: Kiểm tra xem một số có phải là số nguyên tố hay không bằng phép chia thử với độ chính xác 100%

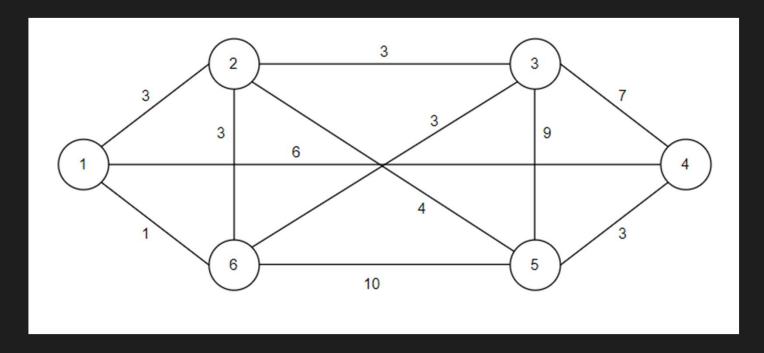
```
function is_prime(x):
for f = 2 to x - 1 do
if x mod f = 0
then return 'not prime'
end for
return 'prime'
```

Độ phức tạp

Theta(n) nếu n là số nguyên tố Theta(ước số nhỏ nhất của n) nếu n không phải số nguyên tố

Bài toán không thích hợp

Cho đồ thị vô hướng, liên thông, có trọng số sau:



Bắt đầu từ đỉnh 1, áp dụng Brute Force, hãy chỉ ra chu trình Hamilton có có đường đi ngắn nhất.

Độ phức tạp

$$\frac{1}{2}(n-1)!$$



THANKS FOR WATCHING!

Huỳnh Nhân Thập - 21521457

Đỗ Bá Huy - 21522137

Nguyễn Tường Duy - 21520782