

# CHỦ ĐỀ: BRUTE FORCE

## NHÓM 5:

*Nguyễn Quốc Huy Hoàng – 20520051*

*Lê Nguyễn Khánh Nam – 20520073*

**Bài 1.** Theo các bạn, các đặc điểm chung của các bài toán Brute Force là gì ?

**Trả lời:**

- Có giới hạn đầu vào nhỏ. Ví dụ: Cho  $n$  điểm trên mặt phẳng, tìm cặp điểm gần nhất ( $n \leq 1000$ ). Dễ thấy ta có thể giải quyết dễ dàng bằng Brute Force bằng cách duyệt mọi cặp rồi kiểm tra trong  $O(N^2)$  mà không cần phải tốn công cài chia để trị cho bài này.
- Cần duyệt qua hết tất cả trường hợp để tính toán. Ví dụ: Mình có một ổ khóa số bị mất mật mã, vì ổ khóa quá tốt nên mình không tìm ra kẻ hở nào để có thể đoán được mật khẩu, lúc này mình cần dùng Brute Force để duyệt qua hết tất cả mật mã của ổ khóa để có thể tìm ra được mật mã đúng.

**Bài 2. Bài toán n hậu ( $n \leq 12$ )**

**Trả lời:**

**Tiếp cận 1,** Giải quyết bài toán theo hướng thuần Brute Force:

- Bước 1: Tạo một mảng  $a$  có kích thước  $n * n$ ,  $a[i][j] = 1$  nếu ở hàng thứ  $i$ , cột thứ  $j$  đã đặt được một quân hậu, ngược lại  $a[i][j] = 0$ . Ban đầu mọi  $a[i][j] = 0$ .
- Bước 2: Sinh mọi dãy tổ hợp  $C(n * n, n)$ , nếu xuất hiện số  $x$  trong dãy thì ô ở vị trí  $((x-1)/n + 1, (x - 1) \% n + 1)$  tồn tại 1 con hậu (tương đương ở trường hợp này  $a[i][j] = 1$ ).
- Bước 3: Với mọi trường hợp của mảng  $a$ , ta kiểm tra xem có thỏa mãn không bằng cách duyệt theo chiều ngang, dọc, chéo. Nếu thỏa mãn thì kết quả cộng một.
- Bước 4: Xuất đáp án và kết thúc chương trình.

**Độ phức tạp**  $O(C(n * n, n))$

## Tiếp cận 2, Giải quyết bài toán theo hướng Brute Force có nhánh cận:

- Bước 1: Tạo các mảng đánh dấu theo cột, theo hàng và theo các đường chéo, nếu giá trị ở ô  $(i, j)$  bằng 1 thì có nghĩa là theo hướng đó đã tồn tại 1 con hậu rồi, ngược lại bằng 0.
- Bước 2: Dùng giải thuật đệ quy bắt đầu duyệt từ ô  $(1, 1)$  và số con hậu đặt được là 0 bằng hàm  $\text{backTrack}(1, 1, 0)$ .
- Bước 3:
  - o Ở ô này ta có một lựa chọn đó là không đặt quân hậu tại đây và đến ô tiếp theo, dùng hàm  $\text{backTrack}(i, j + 1, x)$  (với  $x$  là số quân hậu mà trước đó đã đặt)
  - o Nếu tại vị trí  $(i, j)$  ta có mảng đánh dấu theo cột  $i$ , hàng  $j$  và các đường chéo đi qua ô  $(i, j)$  chưa bị đánh dấu thì ta có thêm 1 lựa chọn đó là đặt 1 quân hậu ở đây và đến ô tiếp theo, ta đánh dấu cột  $i$ , hàng  $j$  và các đường chéo đi qua ô này là 1 và dùng hàm  $\text{backTrack}(i, j + 1, x + 1)$  để đi đến ô tiếp theo. Sau khi vòng đệ quy này kết thúc thì ta phải trả lại giá trị các ô đã đánh dấu lại ban đầu (gán lại giá trị 0).
- 
- Bước 4: Xét hàm đệ quy  $\text{backTrack}(i, j, x)$ 
  - o Nếu  $i \leq n$  và  $j \leq n$  thì ta đi đến bước 3 để xử lý tiếp
  - o Nếu vị trí  $j > n$  tức là đã qua hàng tiếp theo, ta dùng hàm  $\text{backTrack}(i + 1, 1, x)$  và đi đến bước 3.
  - o Nếu  $k == n$  thì tức là đã đặt đủ quân hậu, ta tăng đáp án lên 1 và không đệ quy từ đây nữa.
  - o Nếu vị trí  $i > n$  tức là đã duyệt hết, ta cũng không đệ quy từ đây nữa.
- Bước 5: Xuất đáp án và kết thúc chương trình

**Độ phức tạp:** Vì có nhánh cận nên rất khó tính được độ phức tạp, nhưng vì nó bỏ qua rất nhiều trường hợp đã bị chặn, nên độ phức tạp sẽ tốt hơn thuần Brute Forces rất nhiều.