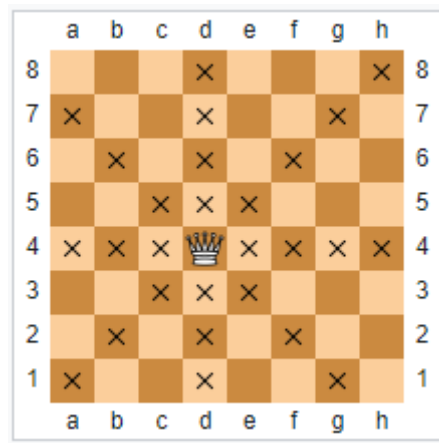


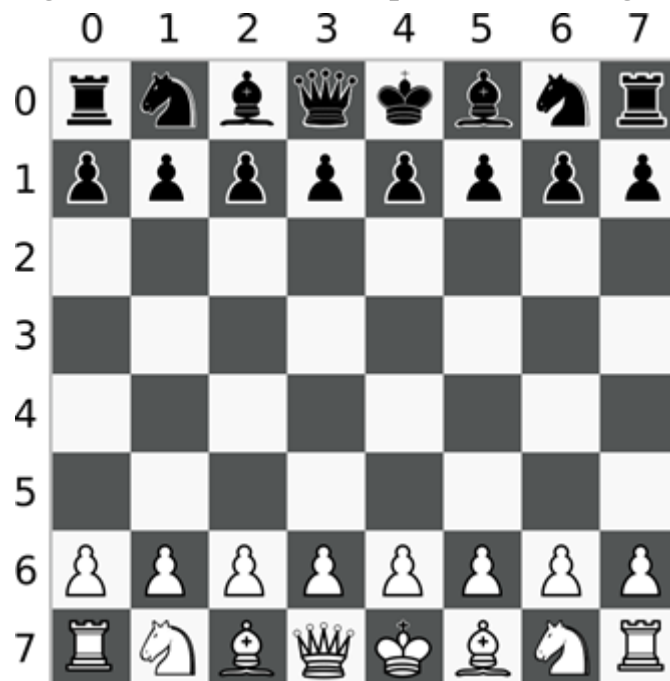
Tư duy tính toán

Tuần 1

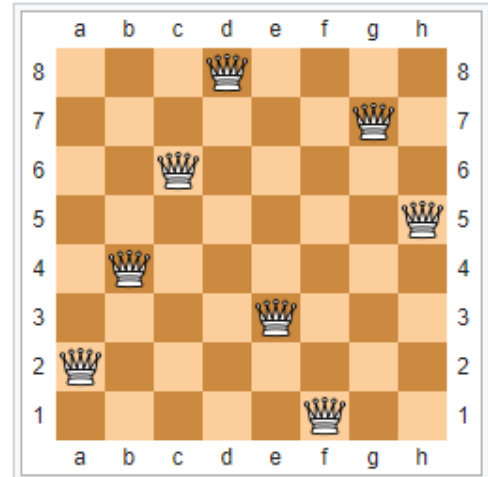
Cho bàn cờ kích thước $N \times N$ và N quân hậu, bài toán xếp hậu là bài toán tìm ra tất cả các cách xếp N quân hậu sao cho không có hai quân hậu nào 'ăn' được nhau. Quân hậu là quân cờ của cờ vua, theo luật quân hậu có thể 'ăn' bất kỳ quân cờ nào theo hàng ngang, cột dọc và hai đường chéo (như hình)



Ta sẽ ký hiệu lời giải của bài toán xếp hậu là một danh sách A gồm N số $A = [a_0, a_1, \dots, a_{n-1}]$, trong đó a_i là vị trí cột của quân hậu ở dòng i (như hình)



Ví dụ: lời giải ở hình bên tương ứng với danh sách [3, 6, 2, 7, 1, 4, 0, 5]



Bài 1. Cho một danh sách số gồm m phần tử $[a_0, a_1, a_{m-1}]$, thực hiện hai yêu cầu sau:

- Đảm bảo rằng $0 \leq a_i < m$, với mọi $i = 0, 1, \dots, m - 1$
- Kiểm tra danh sách số đó có là lời giải cho bài toán xếp hậu.

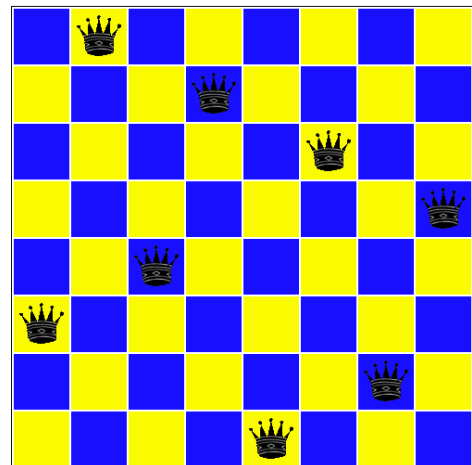
Bài 2. In ra tất cả lời giải cho bài toán xếp hậu khi $N = 7$ bằng cách xét tất cả các trường hợp có thể có khi xếp cờ. Gợi ý: dùng `from itertools import permutations`

Bài 3. Một thuật toán xây dựng lời giải cho bài toán xếp hậu như sau:

1. Tạo danh sách các số lẻ và số chẵn nhỏ hơn N .
2. Nếu N chia 6 dư 2, đổi chỗ 0 và 2 rồi chuyển 4 về cuối danh sách số chẵn.
3. Nếu N chia 6 dư 3, chuyển 1 về cuối danh sách số lẻ và chuyển 0, 2 về cuối danh sách số chẵn.
4. Ghép danh sách số chẵn vào cuối danh sách số lẻ.

Ví dụ: với $N = 8$:

- Số lẻ = [1, 3, 5, 7], số chẵn = [0, 2, 4, 6]
- Vì 8 chia 6 dư 2, nên ta thực hiện bước 2 ở trên, được số chẵn = [2, 0, 6, 4]
- Ghép lại ta được [1, 3, 5, 7, 2, 0, 6, 4]



Viết chương trình python cho thuật toán trên.

Bài 4. Cho danh sách số A được sắp xếp tăng dần và một số k , hãy tìm chỉ số i nhỏ nhất sao cho $k \leq A[i]$. Quy ước: Nếu k lớn hơn số lớn nhất trong A thì $i = \text{len}(A)$.