

Thuật toán tham lam trong bài toán đổi tiền (Medium & Hard)

Thuật toán*:

- Ta nhận thấy, số lượng mệnh giá của các tờ tiền không quá 100, cũng như mệnh giá của mỗi tờ tiền không quá 100.
- Vì vậy, có một cách tham lam đó chính là tách khoản tiền s thành hai phần, c và $s - c$ (c ở đây là một hằng số nào đó, sẽ có giải thích ở phía dưới):
 - Với khoản tiền $s - c$, với $s - c > 0$, ta có thể đổi chúng thành các tờ tiền có mệnh giá lớn nhất. Ngược lại, ta sẽ không thực hiện đổi mà sẽ cho $c = s$ rồi chuyển sang bước tiếp theo.
 - Với khoản tiền c , ta có thể sử dụng những thuật toán chính xác hơn (và đương nhiên sẽ có độ phức tạp lớn hơn) để xử lý. Nhóm mình sẽ sử dụng thuật toán brute force để tìm ra được chính xác số tờ tiền ít nhất có thể đổi. Thuật toán như sau:
 - Xét khoản tiền j ($\forall j = 1..s$), với các tờ tiền từ 1 tới i ($\forall i = 1..n$), thì sẽ xét 2 trường hợp và lấy trường hợp sử dụng số tờ tiền là ít nhất:
 - + Sử dụng tờ tiền i (với điều kiện là $j \geq a_i$, với a_i là mệnh giá của tờ tiền thứ i).
 - + Không sử dụng tờ tiền i .
 - Công thức tổng quát:

$$f(i, j) = \begin{cases} f(i-1, j) & , \quad j < a_i \\ \min \{f(i-1, j), f(i-1, j - a_i)\} & , \quad j \geq a_i \end{cases}$$

- Vì việc chạy đệ quy trên Python khá mất thời gian nên nhóm mình đã sử dụng 2 vòng lặp for để thực hiện thuật toán trên.
- Về hằng số c (với trường hợp $s > c$), ta nên chọn một cách phù hợp để tối ưu khả năng xử lý của thuật toán chính xác trên. Đồng thời, nên chọn c ở mức lớn nhất có thể xử lý để đảm bảo được tính đúng đắn. Trong bài làm của nhóm mình thì nhóm mình chọn $c = 2000$.

(*): Cách làm này có sự tham khảo từ các bài giải sau: <https://oj.vnoi.info/problem/dtdoi/editorial>, tuy nhiên vì không ai đề cập về cách hoạt động của các bài giải này nên nhóm mình chỉ đưa ra cách hiểu của nhóm mình về thuật toán này.