ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÀI TẬP MÔN HỌC
PHÂN TÍCH THIẾT KẾ THUẬT TOÁN

Sinh viên: Đỗ Phương Duy - 23520362

Sinh viên: Nguyễn Nguyên Khang - 22520623

Ngày 1 tháng 12 năm 2024

Mục lục

1 Bài 1: Trò chơi đối kháng

1.1 Đề bài

Cho một số nguyên dương p, hai người chơi (A và B) lần lượt thực hiện các thao tác với p:

- Nếu p là số lẻ, người chơi có thể chọn tăng hoặc giảm p đi 1 đơn vị.
- Nếu p là số chẵn, người chơi bắt buộc giảm p xuống còn $\frac{p}{2}$.
- Người chơi làm cho $p = 0$ sẽ thắng.

Người chơi A luôn đi trước. Hỏi nếu cả hai người chơi đều chơi tối ưu, liệu A có luôn thắng hay không?

1.2 Phân tích bài toán

Đây là một trò chơi thuộc loại **trò chơi đối kháng với tổng không đổi**, trong đó mỗi nước đi của một người chơi đều ảnh hưởng trực tiếp đến trạng thái của người còn lại.

1.3 Các giới hạn và phương pháp giải

1.3.1 Trường hợp $p \leq 10$

1.3.1.1 Ý tưởng: Dùng phương pháp **duyệt trạng thái** để kiểm tra từng giá trị của p. Với mỗi giá trị p, xác định trạng thái thắng/thua:

- Nếu người chơi có thể làm đối thủ rơi vào trạng thái thua, thì trạng thái hiện tại là thắng.
- Nếu mọi nước đi đều dẫn đến trạng thái thắng của đối thủ, trạng thái hiện tại là thua.

1.3.1.2 Mã giả:

```
function canAWin(p):
    if p == 0:
        return False  # Trạng thái thua

    if p is odd:
        return not canAWin(p - 1) or not canAWin(p + 1)  # A thắng nếu B thua
    else:
        return not canAWin(p // 2)  # A thắng nếu B thua khi chia đôi
```

1.3.2 Trường hợp $p \leq 10^6$

1.3.2.1 Ý tưởng: Dùng phương pháp **quy hoạch động** để lưu trữ trạng thái thắng/thua của từng giá trị p trong một mảng dp. Quy tắc chuyển trạng thái:

- Nếu p là lẻ: $dp[p] = \neg dp[p-1] \vee \neg dp[p+1]$.
- Nếu p là chẵn: $dp[p] = \neg dp[p//2]$.

1.3.2.2 Mã giả:

```
function canAWin(p):
    dp = array of size (p+1) initialized to False
    dp[0] = False  # Trạng thái thua

    for i from 1 to p:
        if i is odd:
            dp[i] = not dp[i - 1] or not dp[i + 1]
        else:
            dp[i] = not dp[i // 2]

    return dp[p]
```

1.3.3 Trường hợp $p \leq 10^{18}$

1.3.3.1 Ý tưởng: Với giới hạn lớn, sử dụng **quy luật chiến thắng**:

- Nếu p là số lẻ, người chơi có thể chọn nước đi để ép đối thủ rơi vào trạng thái bất lợi.
- Nếu p là số chẵn, người chơi bắt buộc chia đôi p.

*NOTE: Mình vẫn chưa tìm ra quy luật :«<

1.4 Kết luận

- Với $p \leq 10$: Sử dụng phương pháp duyệt trạng thái.
- Với $p \leq 10^6$: Sử dụng quy hoạch động để tính toán hiệu quả.
- Với $p \leq 10^{18}$: Sử dụng quy luật tối ưu để giải quyết.

2 Bài 2: Trò chơi đồng xu

2.1 Đề bài

Một chồng gồm n đồng xu. Hai người chơi (A và B) lần lượt bốc từ 1 đến k đồng xu ($x \leq k$). Người chơi không thể bốc đồng xu (khi $n = 0$) sẽ thua.

Yêu cầu: Tìm tất cả giá trị k ($k \leq n$) sao cho A luôn thắng nếu cả hai chơi tối ưu.

2.2 Phân tích bài toán

- Đây là **trò chơi có tổng bằng không**, bài toán thuộc loại **trò chơi Grundy**, nơi trạng thái thắng/thua của trò chơi phụ thuộc vào trạng thái còn lại sau mỗi lượt chơi. - Quy luật: - Nếu n là trạng thái mà đối thủ không thể thắng, thì trạng thái đó là thắng cho người chơi hiện tại. - Sử dụng số Grundy để xác định trạng thái của trò chơi.

2.3 Các giới hạn và phương pháp giải

2.3.1 Trường hợp $n \leq 1000$

2.3.1.1 Ý tưởng: Sử dụng **quy hoạch động** để tính trạng thái thắng/thua cho từng n với từng giá trị k. - Quy tắc chuyển trạng thái: - Nếu $n = 0$: $dp[0][k] = \text{False}$ (trạng thái thua). - Nếu $n > 0$: $dp[n][k] = \exists x \, (1 \leq x \leq k \land dp[n - x][k] = \text{False})$.

2.3.1.2 Mã giả:

```
function findWinningKs(n):
    result = []  # Danh sách giá trị k đảm bảo A luôn thắng
    for k in 1 to n:
        dp = array of size (n+1) initialized to False
        dp[0] = False  # Trạng thái thua nếu không còn đồng xu

        for i from 1 to n:
            for x from 1 to min(i, k):  # Xét tất cả số đồng xu có thể bốc
                if not dp[i - x]:
                    dp[i] = True
                    break

        if dp[n]:
            result.append(k)
    return result
```

2.3.2 Trường hợp $n \leq 10^{18}$

2.3.2.1 Ý tưởng: Áp dụng lý thuyết Grundy để xác định trạng thái: - Số Grundy của trạng thái n phụ thuộc vào các trạng thái $n - x$ (với $1 \leq x \leq k$). - Sử dụng định nghĩa số Grundy:

$$G(n) = \text{Mex}(\{G(n-x) \mid 1 \leq x \leq k\})$$

trong đó: - $G(n)$ là số Grundy của trạng thái n. - Mex: Số nguyên nhỏ nhất không thuộc tập hợp.

2.3.2.2 Mã giả:

```
function grundy(n, k):
    if n == 0:
        return 0  # Trạng thái thua
    reachable = set()
    for x from 1 to min(n, k):
        reachable.add(grundy(n - x, k))
    return mex(reachable)  # Giá trị Grundy

function findWinningKs(n):
    result = []
    for k in 1 to n:
        if grundy(n, k) != 0:  # A thắng nếu số Grundy khác 0
            result.append(k)
    return result
```

2.4 Kết luận

- Với $n \leq 1000$: Sử dụng quy hoạch động để tính toán trạng thái thắng/thua.
- Với $n \leq 10^{18}$: Sử dụng lý thuyết Grundy để tối ưu.

Tài liệu