**I. BÀI TOÁN**

*Đề bài:* Chú ếch Freddie muốn di chuyển trên một dòng sông, bắt đầu từ vị trí *0* và kết thúc tại vị trí *n*. Trên dòng sông có *m* bông hoa súng, ở vị trí *0* và *n* chắc chắn có hoa súng. Freddie chỉ có thể di chuyển trên những bông hoa súng này và mỗi lần nhảy chú không thể nhảy quá *r* đơn vị.

*Yêu cầu:* Tìm con đường mà Freddie sẽ nhảy từ *0* đến *n*, sao cho số bước nhảy là ít nhất. Đề bài luôn có đường đi thỏa mãn.

**II. THUẬT TOÁN**

1. **Mô tả thuật toán**

*Dữ liệu vào (Input):*

- Độ dài của dòng sông *n, n* là số nguyên dương*.*

- Khoảng cách nhảy tối đa của chú ếch *r, r* là số nguyên dương*.*

- Số lượng bông hoa súng *m, m* là số nguyên dương, *m ≤ n.*

- *lilys[0..m-1]*, mảng các số nguyên lưu vị trí các bông hoa súng trên dòng sông (*0 ≤ lilys[i] ≤ n, lilys[i] < lilys[i+1], lilys[0] = 0, lilys[m-1] = n*)

*Dữ liệu ra (Output):*

- Đường đi cần ít số bước nhảy nhất mà chú ếch Freddie sẽ thực hiện để di chuyển từ vị trí *0* đến vị trí *n*

*Thuật toán:*

- Gọi *H* là danh sách các bước nhảy mà Freddie sẽ thực hiện (ban đầu danh sách này rỗng)

- Gọi vị trí hiện tại của Freddie là *cur\_position = 0*.

- Thực hiện lặp khi *cur\_position < n*:

+ Tìm vị trí *next\_position* xa nhất mà Freddie có thể nhảy đến từ *cur\_position*.

+ Thêm bước nhảy từ *cur\_position* tới *next\_position* vào danh sách H.

+ Gán *cur\_position = next\_position*.

- Danh sách *H* chính là kết quả bài toán.

1. **Giả mã**

|  |
| --- |
| **algorithm** furthest\_reachable\_lily(cur\_lily, lilys, m, r):  **for** i = cur\_lily["index"]+1 **to** m:  **if** cur\_lily["position"]+r < lilys[i]: **break**  next\_lily = {"index": i, "position": lilys[i]}  **return** next\_lily |
| **algorithm** frog\_hopping(lilys, r, n):  H = [0]  cur\_lily = {"index": 0, "position": 0}  **while** cur\_lily["position"] < n:  next\_lily = furthest\_reachable\_lily(  cur\_lily, lilys, m, r  )  H.append(next\_lily["position"])  cur\_lily = next\_lily  **return** H |

1. **Chứng minh**

Để chứng minh tính đúng đắn của thuật toán trên, ta cần phải chứng minh 2 tính chất sau:

* *Tính hợp lệ:* Thuật toán tìm ra đường đi hợp lệ (thỏa mãn tính chất đề bài).
* *Tính tối ưu:* Thuật toán tìm ra đường đi tối ưu (không có cách di chuyển nào tốt hơn).
  1. **Tính hợp lệ**

Một đường đi hợp lệ là cách di chuyển từ vị trí bắt đầu tới vị trí kết thúc sao cho khoảng cách giữa 2 vị trí liên tiếp không vượt quá *r*.

***Chứng minh phản chứng:***

Giả sử đường đi tìm được thông qua thuật toán trên không thỏa mãn.

Gọi vị trí của các bông hoa súng là *x1 < x2 < … < xm*

Theo giả sử thuật toán trên không tìm được đường đi thỏa mãn nên chú ếch sẽ bị kẹt tại bông hoa súng thứ *k (1 ≤ k < m)*và không thể nhảy được đến bông hoa súng tiếp theo. Nghĩa là chú ếch không thể nhảy được đến bông hoa súng thứ *k+1*, vậy *xk + r < xk+1* ***(1)***

Do đề bài ghi rõ là luôn có đường đi thỏa mãn, nghĩa là bắt buộc phải có bước nhảy nào đó bắt đầu từ trước bông hoa súng thứ *k+1* và nhảy đến bông hoa súng thứ *k+1* hoặc xa hơn nữa. Nói các khác là tồn tại bông hoa súng thứ *s (1 ≤ s ≤ k)* thỏa mãn: *xs + r ≥ xk+1* và do *xk > xs*nên *xk + r > xk+1* ***(2)***

Ta thấy ***(1)*** với ***(2)*** mâu thuẫn với nhau, nên điều giả sử trên là sai. Vậy thuật toán trên luôn tìm được đường đi hợp lệ.

* 1. **Tính tối ưu**

Gọi *H* là đường đi tìm được bởi thuật toán trên và *H\** là đường đi tối ưu bất kì (do có thể có nhiều cách di chuyển tối ưu khác nhau). *|H|* và *|H\*|* lần lượt là số bước nhảy của *H* và *H\**. Dễ nhận thấy *|H| ≥ |H\*|* (do *H\** là cách di chuyển tối ưu nên số bước nhảy trong đó là nhỏ nhất có thể)

Gọi *p(i, H)* là vị trí của chú ếch sau khi thực hiện *i* bước nhảy đầu tiên theo cách di chuyển *H*, ta chứng minh bổ đề sau:

Với mọi *i* thỏa mãn *0 ≤ i ≤ |H\*|* ta có *p(i, H) ≥ p(i, H\*)*.

***Chứng minh quy nạp:***

Với *i = 0*, ta có *p(0, H) = 0 ≥ 0 = p(0, H\*)*, do chú ếch mới chỉ ở vị trí xuất phát chưa di chuyển.

Giả sử bổ đề trên đúng tới *i (0 < i < |H\*|)*, ta cần chứng minh *p(i+1, H) ≥ p(i+1, H\*)*, xét 2 trường hợp:

* ***Trường hợp 1:*** *p(i, H) ≥ p(i+1, H\*)*

Do mỗi bước nhảy đều di chuyển về phía trước nên *p(i+1, H) ≥ p(i, H)* suy ra *p(i+1, H) ≥ p(i+1, H\*)*

* ***Trường hợp 2:*** *p(i, H) < p(i+1, H\*)*

Mỗi bước nhảy không vướt quá *r* đơn vị nên *p(i+1, H\*) ≤ p(i, H\*) + r*.

Ta lại có *p(i, H) ≥ p(i, H\*)* (theo giả sử bổ đề đúng tới *i*) nên *p(i+1, H\*) ≤ p(i, H) + r*, nghĩa là từ vị trí *p(i, H)* có thể nhảy đến vị trí *p(i+1, H\*)*

Do thuật toán của ta sẽ tìm ra *p(i+1, H)* là vị trí xa nhất mà chú ếch có thể nhảy đến từ *p(i, H)* nên *p(i+1, H) ≥ p(i+1, H\*)*

Như vậy bổ đề trên là đúng.

Để chứng minh đường đi ta tìm được theo thuật toán trên là tối ưu, ta cần chứng minh *|H| = |H\*|*

Gọi *k = |H\*|*, theo như bổ đề ta có *p(k, H) ≥ p(k, H\*)*. Vì *k* là bước nhảy cuối cùng theo đường đi *H\** nên *p(k, H\*) = n*, suy ra *p(k, H) ≥ n* ***(3)***

Do thuật toán của ta không nhảy vượt quá *n* nên *p(k, H) ≤ n* ***(4)***

Từ ***(3)*** và ***(4)*** suy ra *p(k, H) = n.*

Theo thuật toán của ta thì thì chú ếch sẽ tới được vị trí thứ *n* sau *k* bước, như vậy *|H| = |H\*|*. Do *|H\*|* là số bước nhảy ít nhất để di chuyển từ *0* đến *n* nên đường đi *H* là tối ưu.

Vậy thuật toán trên sẽ tìm ra lời giải tối ưu.

**III. ĐÁNH GIÁ ĐỘ PHỨC TẠP**

Thời gian nhập dữ liệu vào là *T1 =* *O(m)*

Gọi *k* là số bước nhảy cần thực hiện

Gọi *p1 < p2 < … < pk*là chỉ số của các bông hoa súng trên đường đi.

*ti* là số phép tính để tính toán bước nhảy từ bông hoa súng thứ *i* tới bông hoa súng thứ *i+1*

Ta có với mỗi bông hoa súng *pi*số phép tính cần dung để tìm bông hoa súng tiếp theo là *ti = pi+1 – pi (0 ≤ i < k)*

Như vậy số phép tính cần thực hiện để tìm ra đường đi là:

Thời gian để tìm ra đường đi thỏa mãn là *T2 = O(m)*

Vậy độ phức tạp của thuật toán là: *T1 + T2 = O(max{m, m}) = O(m)*

**IV. XÁC ĐỊNH BỘ DỮ LIỆU**

*Quy ước:*

* *n* là độ dài dòng sông
* *r* là khoảng cách nhảy tối đa
* *m* là số lượng hoa súng trên dòng sông
* *lilys* là vị trí của các bông hoa súng.
* *H* là đường đi chứa các bước nhảy của chú ếch
* *cur\_lily* là vị trí hiện tại của chú ếch
* *next\_lily* là vị trí tiếp theo chú ếch sẽ nhảy tới

1. **Bộ dữ liệu 1**

*Dữ liệu vào (Input):*

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **n** | | | 10 | | | | | | | | |
| **r** | | | 4 | | | | | | | | |
| **m** | | | 6 | | | | | | | | |
| **lilys** | | | 0 1 3 4 6 10 | | | | | | | | |
|  |  |  | |  |  |  |  |  |  |  |  |

*- Khởi tạo:*

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **H** | | {0} | | | | | | | | | | |
| **cur\_lily** | | {"index": 0, "position": 0} | | | | | | | | | | |
|  | |  | |  |  |  |  |  |  |  |  |  |
|  | |  | |  |  |  |  |  |  |  |  |  |
| 0 | | 1 | | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

*- Bước 1:*

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **H** | | {0} | | | | | | | | | | |
| **cur\_lily** | | {"index": 0, "position": 0} | | | | | | | | | | |
| **next\_lily** | | {"index": 3, "position": 4} | | | | | | | | | | |
|  | |  | |  |  |  |  |  |  |  |  |  |
|  | |  | |  |  |  |  |  |  |  |  |  |
| 0 | | 1 | | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

*- Bước 2:*

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **H** | | {0} | | | | | | | | | | |
| **cur\_lily** | | {"index": 3, "position": 4} | | | | | | | | | | |
| **next\_lily** | | {"index": 4, "position": 6} | | | | | | | | | | |
|  | |  | |  |  |  |  |  |  |  |  |  |
|  | |  | |  |  |  |  |  |  |  |  |  |
| 0 | | 1 | | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

*- Bước 3:*

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **H** | | {0} | | | | | | | | | | |
| **cur\_lily** | | {"index": 4, "position": 6} | | | | | | | | | | |
| **next\_lily** | | {"index": 5, "position": 10} | | | | | | | | | | |
|  | |  | |  |  |  |  |  |  |  |  |  |
|  | |  | |  |  |  |  |  |  |  |  |  |
| 0 | | 1 | | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

*- Bước 4 (dừng vòng lặp vì chú ếch đã nhảy tói vị trí n):*

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **H** | | {0} | | | | | | | | | | |
| **cur\_lily** | | {"index": 5, "position": 10} | | | | | | | | | | |
| **next\_lily** | |  | | | | | | | | | | |
|  | |  | |  |  |  |  |  |  |  |  |  |
|  | |  | |  |  |  |  |  |  |  |  |  |
| 0 | | 1 | | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

*Dữ liệu ra (Output)*

- Số bước nhảy cần thực hiện là 3

- Đường đi của chú ếch: 0 → 4 → 6 → 10

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

1. **Bộ dữ liệu 2**