

## Bài tập CS112: Quy hoạch động

Nhóm : 2

Nguyễn Thiện Nhân 23521083

Trần Vinh Khánh 23520726

### Bài tập lý thuyết

#### 1. Có phải mọi bài toán đều có thể giải quyết bằng quy hoạch động không? Tại sao?

Không phải bài toán nào cũng có thể giải quyết bằng quy hoạch động. Tuy rằng quy hoạch động mạnh nhưng không thể bao hàm tất cả.

#### 2. Trong thực tế, bạn đã gặp bài toán nào có thể áp dụng quy hoạch động? Hãy chia sẻ cách tiếp cận.

Trong thực tế mình thấy bài toán quy hoạch động trên đồ thị là có tính trừu tượng và dễ tiếp cận nhất. Bạn đi tới từ một số đỉnh và bạn sẽ lấy kết quả (có thể là max hoặc tổng của các đỉnh được đi tới). Tưởng tượng như vậy làm mình dễ hơn khi học quy hoạch động

#### 3. Hãy phân tích và làm rõ ưu, nhược điểm của 2 phương pháp Top down và Bottom up. Bạn sẽ ưu tiên phương pháp nào? Vì sao?

- Phương pháp Top down :
  - Ưu điểm : dễ hiểu, dễ cài đặt và linh động
  - Nhược điểm: Chi phí thường cao hơn và khó tối ưu hơn
- Phương pháp Bottom up :
  - Ưu điểm : Tốc độ xử lý và chi phí xử lý tốt hơn
  - Nhược điểm: Khó hiểu hơn và khó trong việc cài đặt, linh hoạt

Khi gặp một bài toán mình thường sẽ tư duy theo Top down xuống trước để có thể hiểu được bài toán. Sau đó nếu cảm thấy cần tối ưu thì mình sẽ suy nghĩ thêm và có thể áp dụng phương pháp Bottom up

### Bài tập thực hành

#### Bài 1 : Chú Ếch

##### Ý tưởng cơ bản

Ta thấy rằng ở mỗi bước thì chú ếch chỉ có thể nhảy tối đa 100 bước. Theo tính toán nếu như chúng ta chạy tất cả giả định mà chú ếch có thể nhảy ở mỗi bước thì ta sẽ chỉ mất chi phí  $O(n * k)$ .

Sử dụng mảng  $f[i]$  với ý nghĩa là kết quả tốt nhất khi chú ếch đã có thể nhảy đến  $i$ . Với  $j$  từ khoảng  $i + 1$  đến  $i + k$  ta sẽ xây dựng  $f[j] = \min(f[j], f[i] + |h[j] - h[i]|)$

Thu được kết quả là  $f[n]$

## Mã giả

---

```
1 Def DP()
2
3 //Khởi tạo mảng f[i] với f[1] = 0 và những ô chưa tối là +∞
4   for (int i = 2; i <= n; i++) f[i] = INT_MAX;
5
6   for (int i = 1; i <= n; i++)
7       for (int j = i + 1; j <= min(n, i + k); i++)
8           f[j] = min(f[j], f[i] + abs(h[i] - h[j]));
9
10  return f[n];
11
```

---