Quy hoạch động

1. QUY HOẠCH ĐỘNG ĐẾM CÁCH
2. Công thức dùng đệ quy để giải mọi bài toán Quy hoạch động đếm cách.

* Tìm cách để phân tích bài toán ra thành tổng những bài toán nhỏ hơn tương đương.
* Tìm stopping condition của bài đếm(trường hợp nào return đếm là 1? Trường hợp nào đếm là 0?).
* Nhớ(!): sử dụng bảng nhớ phụ để giúp Đệ quy tránh phải tính toán lại(Time Limit Exceeded)

Big Notes:

* Quay Lui: có phần tính sau phần đệ quy -> phần sau của phần đệ quy đó sẽ được thêm vào stack và quay lui lại tính khi Thread chính chạy hoàn tất.( Chạy hết Stack trong chương trình ).
* Có 2 cách tiếp cận một bài toán quy hoạch động.
  + Top-Down
  + Bottom-Up
* Top-Down sử dụng khi:
  + Tìm ra bài toán nhỏ hơn tương đương(bài toán nhỏ hơn giúp bạn giải bài toàn lớn).
  + Tìm được điều kiện dừng.
  + Có thể sử dụng bảng nhớ phụ để tránh đệ quy lặp nhiều lần.
* Bottom-Up sử dụng khi:
  + Bài toán con có thể giải được ngay -> Giải ra bài toán lớn.
  + Ko cần điều kiện dừng.
* Chỉ có parameters trong hàm mới giữ giá trị của params ở thời điểm đó. Những biến còn lại luôn có giá trị cập nhật tương ứng với luồng chính.

1. QUY HOẠCH ĐỘNG DÙNG ÍT NHẤT

* Công thức dùng Đệ Quy để giải nhiều bài Quy hoạch động DÙNG ÍT NHẤT

1. Tìm cách chẻ bài đếm ra thành tổng những bài nhở hơn tương đương.

* Sử dụng hàm MIN để xây dựng tối ưu của từng điểm nhằm xây dựng tối ưu của toàn bài.

Cách làm: Giá trị tại chỗ + Min(nhánh 1, nhánh 2)

1. Tìm stopping condition của bài đếm (trường hợp nào return đếm là 1? Trường hợp nào đếm là 0?)

* Tùy tính chất bài tập, nhưng nhất quyết phải trả về 2 trường hợp: giá trị kết thúc có trên dữ liệu và giá trị kết thúc ở ngoài biên khiến hàm MIN không chọn giá trị đó(ở đây ta dùng inf).

1. Nhớ (!): sử dụng bảng nhớ phụ để giúp Đệ quy tránh phải tính toán lại (Time Limit Exceeded)