

Bài 2

2.1 Đổi tiền

+ Bài toán yêu cầu số tờ tiền là ít nhất (tối ưu).

+ Ta xây dựng hàm $F(M)$ để tính số lượng tờ tiền ít nhất tạo thành M đơn vị tiền :

$$F(0) = 0$$

$$F(d) = 1 \text{ nếu } d \in D$$

$$F(M) = \min(F(M-d)+1) \text{ với } d \in D$$

+ QHĐ:

$F[j]$ là bảng phương án tính số lượng tờ tiền ít nhất tạo thành j đơn vị tiền :

Cơ sở QHĐ:

$$F[d] = 1 \text{ với } d \in D$$

$$\text{Công thức tính : } F[j] = \min(F[j], F[j-d]+1) \text{ với } d \in D$$

Kết quả $F[M]$ (= 0 nếu không thể tạo thành M đơn vị tiền)

+ Độ phức tạp $O(M*N)$

2.2 Bố trí phòng họp

+ Sắp xếp lại mảng lịch trình, đưa bài toán về dạng bài dãy con tăng dài nhất ta thấy

$F[j]$ là số cuộc họp sẽ được phục vụ nếu cuộc họp cuối bắt đầu lúc a_j và kết thúc lúc b_j .

$$\text{Cơ sở QHĐ : } F[j] = 1 \quad \forall j \in [1, N]$$

$$F[j] = \min(F[j], F[i]+1) \text{ nếu } a_i \leq b_i \leq a_j \leq b_j \text{ với } i \in [1, j]$$

Bài 3

3.2 BANHCHUNG

+ Tham khảo : <http://ntucoder.net/Problem/Details/5518>

+ Phát biểu lại bài toán :

Cho N giá trị nguyên dương A_i , tìm tổng lớn nhất $S \leq M$ với S là tổng của 1 tổ hợp bất kỳ trong A và M là giá trị nguyên dương cho trước.

+ Dễ thấy đây là 1 bài toán QHĐ tìm cách tạo thành 1 tổng M . Tuy nhiên bài toán này chỉ cho phép sử dụng 1 lần A_i mỗi loại nên cần thay đổi lại cách cài đặt sao cho mỗi giá trị A_i chỉ xuất được sử dụng tối đa 1 lần với mỗi tổng được tạo thành.

+ Độ phức tạp $O(NM)$