**Homework 6**

Bài 1:

1. Bài toán tìm dãy con đơn điệu tăng dài nhất

* File code: longest\_increase\_sub.py

1. Bài toán xếp ba lô 0 – 1

* File code: knapsack.py

Bài 2:

1. Bài toán đổi tiền
   1. Phân tích bài toán

* Input: dãy mệnh giá của từng tờ tiền, giá trị cần đổi
* Output: số lượng tờ tiền ít nhất, số tờ tiền mỗi loại.
  1. Giải thuật
* Ta xây dựng mảng có n + 1 phần tử, với n là số tiền cần đổi.
* Giá trị của mảng tại vị trí thứ i : A[i] là số tờ tiền ít nhất cần để đổi được mệnh giá i.
* Dễ nhận thấy: A[i] = 1 với I là các mệnh giá của từng tờ tiền có thể đổi.
* A[j] = min(A[k] + A[j – k]) với k > 0 và k < j
  1. Chương trình
* File code change\_money.py

1. Bài toán bố trí phòng họp
   1. Phân tích bài toán

* Input: Dãy các hoạt động cần lựa chọn.
* Output: Dãy các hoạt dộng sao cho số hoạt động là nhiều nhất
  1. Giải thuật
* Tương thự như bài tập tìm dãy con đơn điệu tăng dài nhất, hoạt động B được coi là lớn hơn A nếu thơi gian bắt đầu của hoạt động B lớn hơn thời gian kết thúc của hoạt động A
  1. Chương trình
* File code: activity\_selector.py

Bài 3:

1. Bài toán: Tính tổ hợp chập k của n
   1. Phân tích

* Để tính tổ hợp chập k của n theo định nghĩa ta cần tính được các giá trị n!, k!, và (n – k)!. Nhưng nếu ta phải tính nhiều lần với các giá trị n và k khác nhau thì cách làm trên sẽ tốn thời gian và khoonbg tận dụng được việc tính các giai thừa lớn từ các gia thừa nhỏ
* Công thức truy hồi:

Chứng minh:

* 1. Thuật toán
* Dựng bảng có số hàng là n + 1 và số cột là k + 1
* Giá trị trong bảng tại hàng i và cột j : T[i][j] là tổ hợp chập j của i, với  
   i >= j. Các giá trị T[i][j] = 0 với i < j do không tồn tại.
* Ta tính các giá trị trong bảng T qua công thức T[1][1] = 1, T[i][1] = i, T[i][j] = T[i – 1][j – 1] + T[i – 1][j]
  1. Chương trình
* File code: combination.py