* **Lưu chỉ số của mảng ban đầu:**

id=[]

for i in range(n):

id.append(i+1)

* **Sắp xếp kèm theo chỉ số:**

def qs (d,c):

i = d

j = c

m = a[(i+j)//2]

while i <= j:

while a[i] < m:

i = i + 1

while a[j] > m:

j = j - 1

if i <= j:

a[i],a[j] = a[j],a[i]

id[i],id[j] = id[j], id[i]

i = i + 1

j = j - 1

if d < j:

qs(d,j)

if i < c:

qs(i,c)

**Ứng dụng:** sắp xếp một đối tượng, ví dụ hs (họ tên, ngày sinh, điểm trung bình) sắp xếp tăng theo điểm trung bình.

* **Tạo file input: tạo file DAY.INP gồm 1000000 số nguyên có giá trị trong phạm vi từ 1 đến 10000**

fout = open("DAY.INP","w")

import random as r

for i in range(1000000):

print(r.choice(range(1,10001)), end = " ",file = fout)

fout.close()

* **Thuật sắp xếp:**
* Nổi bọt: chậm nhất
* Quick sort: chậm nhì
* Đếm phân phối: nhanh nhất, phụ thuộc vào **giá trị của từng phần tử.**
* Hàm có sẵn trong python: nhanh nhì, chỉ áp dụng cho **kiểu dữ liệu mặc định**.
* Chi tiết từng thuật toán, code = python
* Nổi bọt: lần lượt xét dãy từ n giảm 2, mỗi dãy tìm phần tử nhỏ nhất đẩy về đầu dãy

def bbsort():

for i in range(n-1):

for j in range(n,i+1,-1):

if a[j-1] > a[j]:

a[j-1],a[j] = a[j],a[j-1]

* Quick sort: chậm nhì

def qsort(d,c):

i = d

j = c

m = a[(i+j)//2]

while i <= j:

while a[i] < m:

i = i + 1

while a[j] > m:

j = j - 1

if i <= j:

a[i],a[j] = a[j],a[i]

i = i + 1

j = j - 1

if d < j:

qsort(d,j)

if i < c:

qsort(i,c)

* Đếm phân phối: nhanh nhất, phụ thuộc vào giá trị của từng phần tử.

maxai = 10001

#Đếm số lần xuất hiện của giá trị **a[i]**

c = [0]\*maxai

for i in range(n+1):

c[a[i]] += 1

#Phân phối

b = []

for i in range(maxai):

while c[i] != 0:

b.append(i)

c[i] -= 1

* Hàm có sẵn trong python: nhanh nhì, chỉ áp dụng cho **kiểu dữ liệu mặc định**.

a.sort()

**Giá trị nhỏ thứ k MINK**

Cho dãy a1, a2,…, an, các số đôi một khác nhau và số nguyên dương k (1 < k < n). Hãy đưa ra giá trị, vị trí nhỏ thứ k trong dãy.

***Dữ liệu vào:*** đọc từ tệp văn bản **MINK.INP** gồm:

* Dòng đầu chứa dãy a1, a2,…, an. Các số cách nhau bởi dấu cách.
* Dòng thứ hai chứa số nguyên dương k.

***Kết quả:*** lưu vào tệp **MINK.OUT** gồm: giá trị và vị trí của phần tử nhỏ thứ k trong dãy a. Hai số cách nhau một dấu cách.

*Ví dụ:*

|  |  |
| --- | --- |
| **MINK.INP** | **MINK.OUT** |
| 10 4 2 23 6 21 9 8 15 7  5 | 8 8 |

**Hướng dẫn:**

- Đọc dữ liệu từ tệp MINK.INP lưu vào list a chứa các số nguyên, số nguyên k.)

- Tạo mảng id để lưu chỉ số ban đầu của các phần tử.

- Sắp xếp a tăng dần, kèm theo chỉ số id.

- Ghi kết quả là a[k], id[k] ra tệp MINK.OUT.

**Code:**

**Code:**

fin = open("MINK.INP","r")

fout = open("MINK.OUT","w")

li = fin.readline()

a = []

ls = li.split()

for i in ls:

a.append(int(i))

li = fin.readline()

k = int(li)

n = len(a)

print(a)

id=[]

for i in range(n):

id.append(i+1)

def qs (d,c):

i = d

j = c

m = a[(i+j)//2]

while i <= j:

while a[i] < m:

i = i + 1

while a[j] > m:

j = j - 1

if i <= j:

a[i],a[j] = a[j],a[i]

id[i],id[j] = id[j], id[i]

i = i + 1

j = j - 1

if d < j:

qs(d,j)

if i < c:

qs(i,c)

qs(0,n-1)

print(a[k-1],id[k-1],file = fout)

fin.close()

fout.close()

**---------------------------------------**

**Dãy số F DAYF**

Xét dãy F gồm n (2 < n < 106) số nguyên F = (f0, f1,…, fn) định nghĩa:

* fi = 1, khi 0 ≤ i < 2
* fi = (fi-1 + fi-2) mod 128, khi 2 ≤ i ≤ n

Nếu sắp xếp dãy F theo thứ tự không giảm thì số thứ k (k < n) có giá trị bao nhiêu?

***Dữ liệu vào:*** đọc từ tệp văn bản **DAYF.INP** gồm: hai số nguyên dương n và k được phân cách nhau bởi dấu cách.

***Kết quả:*** lưu vào tệp **DAYF.OUT** gồm: giá trị của số thứ k trong dãy F.

*Ví dụ:*

|  |  |
| --- | --- |
| **DAYF.INP** | **DAYF.OUT** |
| 10 7 | 21 |
| 50 28 | 64 |

**Hướng dẫn:**

* Đọc dữ liệu vào từ file DAYF.INP lưu vào 2 biến n, k.
* Tạo dãy f theo định nghĩa của đề bài.
* Ta nhận thấy dãy f có giá trị nguyên và 0 ≤ fi ≤ 127, nên ta sử dụng kĩ thuật đếm phân phối. Tạo dãy c[0], c[1],…c[127], trong đó c[v] là số lần xuất hiện giá trị v trong dãy f.
* Giá trị thứ k của dãy f sau khi sắp xếp là giá trị p nhỏ nhất thoả mãn: c[0]+c[1]+…+c[p]>k.
* Ghi giá trị p ra tệp DAYF.OUT.

**Code:**

* Đọc dữ liệu vào từ file DAYF.INP lưu vào 2 biến n, k.

fin = open("DAYF.INP","r")

fout = open("DAYF.OUT","w")

li = fin.readline()

ls = li.split()

n,k = int(ls[0]),int(ls[1])

* Tạo dãy f theo định nghĩa của đề bài.

f = []

f.append(1)

f.append(1)

for i in range(2,n+1):

f.append((f[i-1]+f[i-2])%128)

* Ta nhận thấy dãy f có giá trị nguyên và 0 ≤ fi ≤ 127, nên ta sử dụng kĩ thuật đếm phân phối. Tạo dãy c[0], c[1],…c[127], trong đó c[v] là số lần xuất hiện giá trị v trong dãy f.

c = [0]\*127

for i in range(n+1):

c[f[i]] += 1

* Giá trị thứ k của dãy f sau khi sắp xếp là giá trị p nhỏ nhất thoả mãn: c[0]+c[1]+…+c[p]>k.

s = 0

kq = 0

for p in range(128):

s = s + c[p]

if s > k:

kq = p

break

* Ghi giá trị p ra tệp DAYF.OUT.

print(kq, file = fout)

fin.close()

fout.close()

* **Ghi chú khi code:**
* Chú ý chỉ số.