**Tham lam**

**Thuật toán quay lui**

**Thuật toán nhánh cận**

**Chia để trị**

**Quy hoạt động**

**Đồ thị**

* **Tính độ phước tạp của thuật toán**

Thuật toán tối ưu là thuật sử dụng ít phép tính, ít tốn bộ nhớ.

Độ lớn dữ liệu là n:

Duyệt từ 1 đến n: O(n)

Duyệt 2 for của n: O(n2)

Duyệt 3 vòng lặp lồng nhau của n: O(n3)

Hai vòng lặp lồng nhau + 1 Vòng lặp thực hiện hiện sau

O(n2) + O(n) + O(n2) = O(n2)

* **Tính trọng lượng của dãy**

Mảng a

Tính F, với Fi = F1+F2+…+Fi

Fi gọi là trọng lượng của dãy con i

**Bài 1. Tích lớn nhất TICHMAX**

Cho một dãy gồm N số nguyên. Hãy tìm ba số trong dãy với tích T của chúng là lớn nhất.

***Dữ liệu vào:*** đọc từ tệp **TICHMAX.INP:**

* Dòng đầu tiên: ghi số N (3≤N≤10000).
* Dòng thứ hai chứa N số nguyên có giá trị tuyệt đối không vượt quá 30000, các số cách nhau một dấu cách.

***Dữ liệu ra:*** ghi ra tệp **TICHMAX.OUT** số T.

***Ví dụ:***

|  |  |
| --- | --- |
| **TICHMAX.INP** | **TICHMAX.OUT** |
| 9  3 5 1 7 9 0 9 -3 10 | 810 |

**Hướng dẫn:**

Tích lớn nhất của 3 phần tử có các trường hợp sau:

* Ba số dương: ba số này phải là **ba số lớn nhất, nhì, ba trong dãy**.
* Một số âm và hai số dương: dãy phải gồm đúng hai số dương, vì nếu không ta có thể lấy tích của ba số dương để đạt giá trị lớn hơn. Ta cũng suy ra ba số này phải là **ba số lớn nhất, nhì, ba trong dãy**.
* Hai số âm và một số dương: số dương phải là số lớn nhất và hai số âm phải là **hai số nhỏ nhất, nhì trong dãy.**
* Ba số âm: dãy phải gồm toàn số âm, vì nếu có một số dương ta cũng có thể lấy tích của số dương đó và hai số âm để thu được tích dương. Ta cũng suy ra được ba số cần tìm phải là **ba số lớn nhất, nhì, ba trong dãy**.

Vậy suy ra T = max(max1\*max2\*max3, min1\*min2\*max1), với max1, max2, max3, min1, min2 lần lượt là số lớn nhất, nhì, ba và số nhỏ nhất, nhì của dãy.

**Code:**

* Đọc dữ liệu

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

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

li = fin.readline()

n = int(li)

a = []

li = fin.readline()

ls = li.split()

for i in range(n):

a.append(int(ls[i]))

- Sắp xếp và tìm kết quả

kq = 0

if n == 3:

kq = a[0]\*a[1]\*a[2]

else:

a.sort()

if n == 4:

max1,max2,max3,min1,min2 = a[n-1],a[n-2],a[n-3],a[0],1

else:

max1,max2,max3,min1,min2 = a[n-1],a[n-2],a[n-3],a[0],a[1]

if max1\*max2\*max3 > min1\*min2\*max1:

kq = max1\*max2\*max3

else:

kq = min1\*min2\*max1

- Sắp xếp và tìm kết quả

print(kq, file = fout)

fin.close()

fout.close()

**Bài 2. Dãy số DAYSO**

Cho một dãy số nguyên a1, a2,…,an. Số ap (1≤p≤n) được gọi là một số trung bình cộng trong dãy nếu tồn tại ba chỉ số i, j, k (i,j,k≤n) đôi một khác nhau sao cho ap = (ai+aj+ak)/3.

***Yêu cầu:*** cho n và dãy a1, a2,…,an. Hãy tìm số lượng các số trung bình cộng trong dãy.

***Dữ liệu vào:*** đọc từ tệp **DAYSO.INP:**

* Dòng đầu ghi số nguyên dương n (3≤n≤1000).
* Dòng thứ hai ghi n số nguyên ai (ai <108), các số cách nhau một dấu cách.

***Dữ liệu ra:*** ghi ra tệp **DAYSO.OUT** số lượng các số trung bình cộng trong dãy.

***Ví dụ:***

|  |  |
| --- | --- |
| **DAYSO.INP** | **DAYSO.OUT** |
| 5  4 3 6 3 5 | 2 |

**Hướng dẫn:**

Sử dụng quay lui (chậm)

Tham lam < O(n3)

* B1: Đọc dữ liệu từ tệp DAYSO.INP lưu vào n, mảng a.
* B2: Sắp xếp a tăng dần.
* B3: Tạo mảng s là tất cả các loại tổng ai + aj với i<j. Tạo mảng id để lưu chỉ số i, j ban đầu của mảng a.
* B4: Sắp xếp mảng s tăng dần kèm theo chỉ số i, j.
* B5: Đếm số lượng, với một số ai bất kì ta kiểm tra xem nó có phải là số trung bình hay không.
* B6: Ghi kết quả ra tệp DAYSO.OUT.

**Code:**

* B1: Đọc dữ liệu từ tệp DAYSO.INP lưu vào n, mảng a.

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

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

li = fin.readline()

n = int(li)

a = []

li = fin.readline()

ls = li.split()

for i in range(n):

a.append(int(ls[i]))

* B2: Sắp xếp a tăng dần.

a.sort()

* B3: Tạo mảng s là tất cả các loại tổng ai + aj với i<j. Tạo mảng id để lưu chỉ số i, j ban đầu của mảng a.

s = []

id = []

for i in range(n-1):

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

s.append(a[i]+a[j])

id.append([i,j])

* B4: Sắp xếp mảng s tăng dần kèm theo chỉ số i, j.

def qs (d,c):

i = d

j = c

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

while i <= j:

while (s[i] < m):

i = i + 1

while (s[j] > m):

j = j - 1

if i <= j:

s[i],s[j] = s[j],s[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)

m = len(s)

qs(0,m-1)

* B5: Đếm số lượng, với một số ai bất kì ta kiểm tra xem nó có phải là số trung bình hay không.

kq = 0

for p in range(n): #kt a[p] la so tbc

t = a[p]

t = t\*3

g = m-1

for k in range(n): #kt a[k] co tao thanh bo (i,j,k)

for q in range(g,0,-1):

if s[q] + a[k] <= t:

break

g = q

if (s[q] + a[k] == t) and (k != id[q][0]) and (k != id[q][1]):

kq += 1

break

* B6: Ghi kết quả ra tệp DAYSO.OUT.

print(kq, file = fout)

fin.close()

fout.close()

**Bài 3. ESEQ**

Cho dãy số nguyên A gồm N phần tử A1, A2,…,AN. Tìm cặp chỉ số i, j thoả mãn:

A1 + A2, +…, + Ai = Aj + Aj+1 + …,+ AN với 1≤i<j≤N.

***Dữ liệu vào:*** đọc từ tệp **ESEQ.INP:**

* Dòng đầu là số nguyên dương N (2≤N≤105).
* Dòng thứ tiếp theo chứa N số nguyên A1, A2,…,AN (|Ai| < 109), các số cách nhau một dấu cách.

***Dữ liệu ra:*** ghi ra tệp **ESEQ.OUT** gồm 1 số là số cặp tìm được.

***Ví dụ:***

|  |  |
| --- | --- |
| **ESEQ.INP** | **ESEQ.OUT** |
| 3  1 0 1 | 3 |

**Hướng dẫn:**

Ta có Fi = A1 + A2 + … + Ai

Aj + Aj+1 + … + AN = Fj = FN – Fj-1

Theo đề bài ta phải tìm cặp i, j thoả mãn Fi = Fj -> F­i = FN – Fj-1 -> Fi + Fj-1 = FN

Bài toán trở thành cho dãy F1, F2, …, FN. Tìm hai phần tử i, x (x=j-1) sao cho Fi + Fx = FN

-B1: Đọc dữ liệu từ tệp ESEQ.INP vào n, mảng a

-B2: Tạo mảng f, trong đó f[i] = a[1]+a[2]+…+a[i]

-B3: Tìm các cặp chỉ số

-B4: In kết quả ra tệp ESEQ.OUT

Ví dụ minh hoạ:

A: 1 2 3 4 5 6 7

F: 1 3 6 10 15 21 28

F4 = 10

F7 = 28

F5 = F7 – F4 = 28 – 10 = 18

**Code:**

* B1: Đọc dữ liệu từ tệp ESEQ.INP vào n, mảng a

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

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

li = fin.readline()

n = int(li)

a = []

li = fin.readline()

ls = li.split()

for i in range(n):

a.append(int(ls[i]))

-B2: Tạo mảng f, trong đó f[i] = a[1]+a[2]+…+a[i]

f = []

s = a[0]

f.append(s)

for i in range(1,n):

s = s + a[i]

f.append(s)

- B3: Tìm các cặp chỉ số

d = 0

for i in range(n-1):

for x in range(i,n):

if f[i] + f[x] == f[n-1]:

d += 1

* B4: In kết quả ra tệp ESEQ.OUT

print(d,file = fout)

fin.close()

fout.close()