**Algorithm Design Methods**

|  |  |
| --- | --- |
| A person in a suit standing outside  Description automatically generated | Lecturer  PGS TSKH Nguyễn Xuân Huy  Viện Công nghệ Thông tin  Viện Hàn lâm Khoa học  và Công nghệ Việt nam  [nxhuy564@gmail.com](mailto:nxhuy564@gmail.com)  0903203800  Connection: Zalo |

Nguyễn Xuân Huy – ID: **0903203800** - Pass: **huynx**

1. Mon July 8 2024

2. Thu July 11 2024

3. Mon July 15 2024

4. Thu July 18 2024

5. Mon July 22 2024

6. Thu July 25 2024

7. Mon July 29 2024

8. Thu July Aug 1 2024

9. Mon Aug 5 2024

10. Thu Aug 8 2024

**Leson 4: Dynamic Programming**

4. Thu July 18 2024

***Some little problems***

### P1. Quadratic equations

Program

# Quadratic Equation

def Go(msg = ' ? '):

if input(msg) == '.': exit(0)

# return n, x1, x2

def QE(a, b, c): # ax^2 + bx + c = 0

delta = b\*b - 4\*a\*c

if delta < 0: return 0, 0.0, 0.0

aa = a + a # 2a

if delta == 0:

x = -b // aa

return 1, x, x

# delta > 0

sd = delta\*\*0.5 # can delta

x1 = (-b+sd) // aa

x2 = (-b-sd) // aa

return 2, x1, x2

# APPLICATION

"""

Test

(x+2)^2 = x^2 + 4x + 4 = 0

a b c = 1 4 4

(x-2)(x+3) = x^2 - x - 6 = 0

a b c = 1 -1 -6

"""

while True:

s = input('a b c or [.] to stop : ')

if s == '.': break

a,b,c = map(int, s.split())

n,x1,x2 = QE(a, b, c)

if n == 0: print('No solutions'); continue

if n == 1: print('Duple solutions', x1, x2); continue

print('Two solutions', x1, x2)

print(' T h e E n d')

### P2. Triangular Number

*Với mỗi số tự nhiên n, số T(n) = 1 + 2 + ... + n được gọi là số tam giác. Dưới đây là sáu số tam giác đầu tiên:*

T(0) = 0

T(1) = 1

T(2) = 1 + 2 = 3

T(3) = 1 + 2 + 3 = 6

T(4) = 1 + 2 + 3 + 4 = 10

T(5) = 1 + 2 + 3 + 4 + 5 = 15

*Nhập các số tự nhiên x từ bàn phím và hiển thị YES nếu x là số tam giác; ngược lại hiển thị NO.*

Algorithm

T(n) = n(n+1)//2 (Gauss)

x = T(n) = n(n+1) // 2

2x = n(n+1)

Can(2x) = n ? n\*(n+1) = 2x

Program

# Triangular Numbers

def Go(msg = ' ? '):

if input(msg) == '.': exit(0)

def T(n): return n\*(n+1)//2

# x = T(n) = n(n+1)// 2

# 2x = n(n+1)

def IsT(x):

if x < 0: return False

xx = x+x

n = int(xx\*\*0.5)

return n\*(n+1) == xx

def Test1(limit):

for n in range(limit):

x = T(n)

print('Is triangular ', x, end = ' ? ')

if (IsT(x)): print(' YES.')

else: print(' NO.')

x += 3

print('Is triangular ', x, end = ' ? ')

if (IsT(x)): print(' YES.')

else: print(' NO.')

def Test2():

while True:

s = input('get a number or [.] to stop: ')

if s == '.': break

if IsT(int(s)): print(' YES.')

else: print(' NO.')

# APPLICATION

Test1(20)

Test2()

print(' T h e E n d')

### P3. Pentagonal Number

*Với mỗi số tự nhiên n, số được gọi là số ngũ giác. Dưới đây là sáu số ngũ giác đầu tiên:*

P(0) = 0

P(1) = 1

P(2) = 5

P(3) = 12

P(4) = 22

P(5) = 35

*Viết chương trình hiển thị mỗi số nguyên không âm cho trước kèm theo thông báo YES nếu số đó là số ngũ giác; ngược lại thì thông báo NO. Nếu gặp số âm thì dừng chương trình.*

Algorithm

Nếu *x = 0* thì *x* là số ngũ giác đầu tiên*.*  Giả sử *x >* 0, khi đó *x* là số ngũ giác thứ *n*, *x* = *P*(*n*) khi và chỉ khi , tức là tam thức bậc hai có ít nhất một nghiệm là số tự nhiên *n*. Ta có , suy ra phải là một số tự nhiên.

Hàm PNum(x) return n nếu x là số ngũ giác thứ n, ngược lại, hàm return -1.

Phương án 1 cần vài phép tính số học.

# Phuong an 1

def PNum1(x):

if x == 0:

return 0

if x < 0:

return -1

n = (1 + int(1+24\*x)\*\*0.5) // 6

return n if P(n) == x else -1

Do *P*(*n*) là dãy tăng chặt nên ta cũng có thể dùng công thức đệ quy để sinh lại dãy *P*.

Ta có,

Vậy

Phương án 2 (cần lần lặp)

# Phuong an 2

def PNum2(x):

if x == 0:

return 0

y, d, n = 0, -2, 0

while y < x:

d += 3

y += d

n += 1

return n if y == x else -1

Program

# Pentagonal Number

def Go(msg = ' ? '):

if input(msg) == '.': exit(0)

def P(n): return n\*(3\*n-1)//2

# x = P(n) = n(3n-1)// 2

# 2x = n(3n-1): 3n^2 - n - 2x = 0 Quadratic EquationD

# a = 3, b = -1, c = -2x, n1, n2 ? 1 nghiem nguyen

# return n: P(n) = x

def IsP(x):

if x < 0: return -1

if x == 0: return 0

sd = (1+24\*x)\*\*0.5

n = int((1+sd)//6)

return n if P(n) == x else -1

def Test1(limit):

for n in range(limit):

x = P(n)

id = IsP(x)

print('Is pentagonal number ', x, end = ' ? ')

if id != -1: print(' YES (' ,id,')')

else: prit(' NO.')

x += 3

id = IsP(x)

print('Is pentagonal number ', x, end = ' ? ')

if id != -1: print(' YES (' ,id,')')

else: print(' NO.')

def Test2():

while True:

s = input('get a number or [.] to stop: ')

if s == '.': break

x = int(s)

id = IsP(x)

if id != -1: print(' YES (' ,id,')')

else: print(' NO.')

# APPLICATION

Test1(20)

Test2()

print(' T h e E n d')

### P4. Hexagonal Number

*Với mỗi số tự nhiên n, số H được gọi là số lục giác. Dưới đây là sáu số lục giác đầu tiên:*

H(0) = 0

H(1) = 1

H(2) = 6

H(3) = 15

H(4) = 28

H(5) = 45

*Viết chương trình đọc từng số nguyên x từ input file NUM.INP và hiển thị YES nếu x là số lục giác; ngược lại hiển thị NO. Nếu x < 0 thì dừng chương trình*

**Leson 3: Searching**

3. Mon July 15 2024

|  |  |
| --- | --- |
| Newton và cuộc cách mạng trong khoa học | *Những bài toán nhỏ*  *đã dẫn tôi đến*  *những bước tiến lớn*  *Isac Newton* (1642-1727) |

***Các bài toán nhỏ vài dòng lệnh***

*P1. Nhập dãy số nguyên, tính tổng các số này.*

*P2. Nhập dãy số nguyên, tính tổng bình phương các số này*

*P3. Kiểm định định lý: tổng n số tự nhiên lẻ bằng*

*Ví dụ,*

*P4. Nhập dãy số nguyên tùy ý, tách thành hai dãy con chẵn và lẻ.*

*P5. Nhập dãy tùy ý, kiểm tra dãy có là palindrome không ?*

*P6. Nhập dãy số nguyên tùy ý, kiểm tra dáng điệu của dãy?*

*1. đồng nhất : mọi phần tử bằng nhau: 3 3 3 … 3*

*2. tăng chặt: 3 7 10 17….*

*3. đồng biến (tăng): 3 3 4 4 7 10 10 …*

*4. giảm chặt: 17 10 7 4…*

*5. nghịch biến (giảm): 10 10 7 7 4 1 -3*

*0. nhấp nhô: 1 7 2 10 4…*

***Rhythm***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | g | t | b | **r (bin)** |  | b = t = g = 0  for i in range(1,len(d)):  if d[i] == d[i-1]:  b = 1; continue  if d[i] > d[i-1]:  t = 1; continue  else: g = 1  r = (4\*g + 2\*t + b) % 7 % 6 |
| 0 | 0 | 0 | **0** | **?** |
| 0 | 0 | 1 | **1** | **=** |
| 0 | 1 | 0 | **2** | **↑** |
| 0 | 1 | 1 | **3** | **↑=** |
| 1 | 0 | 0 | **4** | **↓** |
| 1 | 0 | 1 | **5** | **↓=** |
| 1 | 1 | 0 | **6 🡪 0** | **~** |
| 1 | 1 | 1 | **7 🡪 0** | **~** |

*Program*

"""

Các bài toán nhỏ vài dòng lệnh

P1. Nhập dãy số nguyên, tính tổng các số này.

P2. Nhập dãy số nguyên, tính tổng bình phương các số này

P3. Kiểm định định lý: tổng n số tự nhiên lẻ bằng n^2;

Ví dụ, 1+3+5+7= 4^2=16.

P4. Nhập dãy số nguyên tùy ý, tách thành hai dãy con chẵn và lẻ.

P5. Nhập dãy tùy ý, kiểm tra dãy có là palindrome không ?

P6. Nhập dãy số nguyên tùy ý, kiểm tra dáng điệu của dãy?

1. đồng nhất : mọi phần tử bằng nhau: 3 3 3 … 3

2. tăng chặt: 3 7 10 17….

3. đồng biến (tăng): 3 3 4 4 7 10 10 …

4. giảm chặt: 17 10 7 4…

5. nghịch biến (giảm): 10 10 7 7 4 1 -3

0. nhấp nhô: 1 7 2 10 4…

"""

def Go(msg = ' ? '):

if input(msg) == '.': exit(0)

def GetNum(msg = ' Input a number: '):

return int(input(msg))

def Sum(msg = 'Input: '): # P1

return sum(list(map(int, input(msg).split())))

def SumSquare(msg = 'Input: '): # P2 `

'''

d = list(map(int, input(msg).split()))

print(d)

v = list(map(lambda x: x\*x ,d))

print(sum(v))

'''

print(sum(list(map(lambda x: x\*x, \

list(map(int, input(msg).split()))\

))))

def Theorem(maxn): # P3

for n in range(1, maxn):

print('Test for n = ', n, end = ': ')

d = list(range(1, n+1, 2))

m = len(d)

s = sum(d)

print(' sum ', d, ' =', s, end = ': ')

if s == m\*m: print('YES.')

else: print('NO.')

def SplitEvenOdd(msg = 'Input: '): # P4

# d = list(map(int, input(msg).split()))

# print(d)

even, odd = [], []

for x in list(map(int, input(msg).split())):

if x % 2 == 0: even.append(x)

else: odd.append(x)

print(even, odd)

def Palindrome(msg = 'Input: '): # P5

s = input(msg)

print('YES.') if s == s[::-1] else print('NO.')

def Rhythm(msg = 'Input numbers: '): # P6

d = list(map(int, input(msg).split()))

# print(d)

b = t = g = 0

for i in range(1,len(d)):

if d[i] == d[i-1]: b = 1; continue

if d[i] > d[i-1]: t = 1; continue

else: g = 1

print((4\*g + 2\*t + b) % 7 % 6)

# APPLICATION

# print(Sum('Sum of numbers: '))

# print(SumSquare('Sum square of numbers: '))

# Theorem(20)

# SplitEvenOdd('Input numbers: ')

while True:

# Sum()

# SumSquare()

# Theprem(20)

# SplitEvenOdd()

# Palindrome()

Rhythm()

Go('Press dot key [.] to stop: ')

print(' T h e E n d')

***Blind Searching***

**password** xxxx, x : 0..9

1. Brute Force: tan cong toan dien = vet can

k = 0..9999 ?

for k = 0..9999

Tach k thanh 4 chu so abcd

Kiem tra

2. Biet them thong tin: nam sinh, duoi 24 tuoi

k + y = 2024, k = 2024-y k = 2020…2024 ?

for k = 2020..9999

Tach k thanh 4 chu so abcd

Kiem tra

3. Biet them thong tin: sinh nhat ddmm

dd: 01..31, mm = 01..12 or

mm: 01..12, dd = 01..31

r

for k = 2020..9999

Tach k thanh 4 chu so abcd

Kiem tra

One eye searching: Backtracking: Vet, ko di lai duong da qua

**Leson 2**

2. Thu July 11 2024

Vi sao Next using for i in range(len(s)-2,-2,-1) ?

Go()?

A cartoon of a group of kids

Description automatically generated

'''

Nobita

Doraemon

Suka

Seko

Chaen

'''

def Go(msg = ' ? '):

if input(msg) == '.': exit(0)

def GetNum(msg = ' Input a number: '):

x = int(input(msg))

return x

# APPLICATION

x = GetNum()

print(x)

x = GetNum('Please, input a number: ')

print(x)

x = GetNum('Draemon say: Bạn nào cho tớ một số để tớ làm ảo thuật: ')

print(x)

x = GetNum('Nobita say: Cho tớ một số: ')

print(x)

x = GetNum('Suka say: Cậu nạp một số nhé: ')

print(x)

x = GetNum('Chaen say: Chúng mày nộp cho tao một số ngay: ')

print(x)

x = GetNum('Seko mỏ nhọn say: Tao thèm nhìn thấy một số quá: ')

print(x)

print(' T h e E n d')

**Leson 1: Introduction**

1. Mon July 8 2024

Some recommendations

* Clean Room: One line only + Test
* 5E : five examples