# Greedy Method

*Phương pháp tham* gợi ý chúng ta tìm một *trình tự hợp lí* để duyệt dữ liệu nhằm đạt được mục tiêu một cách chắc chắn và nhanh chóng. Thông thường, dữ liệu được duyệt theo một trong hai trật tự là tăng hoặc giảm dần theo một chỉ tiêu nào đó. Một số bài toán đòi hỏi những dạng thức cải biên của hai dạng nói trên.

## Packing

*Australia*

Anh David muốn xếp các bao hàng lên xe có tải trọng w. Có n bao hàng và biết trọng lượng mỗi bao. Anh muốn chở trọng lượng tối đa với số lượng bao hàng ít nhất. Theo ý bạn David nên làm thé nào?

Giới hạn: n < 100, w < 1000 kg.

|  |  |  |
| --- | --- | --- |
| PACKING.INP | Giải thích | OUTPUT |
| 200 10  12 40 30 25 37  10 22 17 14 14 | Tải trọng xe 200, 10 bao hàng  Trọng lượng mỗi bao hàng  Output:  Tổng trọng lượng (199), số bao (8) | 199 8 |

Thuật toán

Bước 1. Đọc dữ liệu rồi sắp giảm. Độ phức tạp n log n.

Bước 2. Lấy lần lượt các bao hàng từ nặng trở xuống đến khi đạt xấp xỉ tải trọng thì dừng.

Chương trình

# Packing

def Go():

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

def ReadInput():

global w, n, a

with open("PACKING.INP") as f:

# đọc toàn bộ dữ liệu

b = list(map(int,f.read(). split()))

w, n = b[:2]

print(' Tải trong xe: ', w, ' Số bao hàng: ', n)

a = b[2:]

print(' Các bao hàng a: ', a)

def Packing():

# sắp giảm

a.sort(reverse = True)

print(' Sắp giảm a: ', a)

s = 0 # sum

for i in range(n):

s += a[i]

if s > w:

s -= a[i]

break

print(' Result:')

print(' . Tổng trọng lượng xe:', s)

print(' . Só bao: ', i)

def Run():

ReadInput()

Packing()

# APPLICATION

Run()

print(' T h e E n d')

Output

Tải trong xe: 200 Số bao hàng: 10

Các bao hàng a: [12, 40, 30, 25, 37, 10, 22, 17, 14, 14]

Sắp giảm a: [40, 37, 30, 25, 22, 17, 14, 14, 12, 10]

Result:

. Tổng trọng lượng xe: 199

. Só bao: 8

T h e E n d

# Luyện tập Phương pháp tham

## Music box

*Người ta cần ghi N bài hát được mã số từ 1 đến N vào một băng nhạc có thời lượng tính theo phút đủ chứa toàn bộ các bài đã cho. Với mỗi bài hát ta biết thời lượng phát của bài đó. Băng sẽ được lắp vào một máy phát nhạc kiểu cổ đặt tại một phố đi bộ. Khách hàng muốn nghe bài hát nào chỉ việc nhấn phím ứng với bài đó. Để tìm và phát bài thứ i trên băng, máy xuất phát từ đầu cuộn băng, quay băng để bỏ qua i – 1 bài ghi trước bài đó. Tiếp đến, máy sẽ phát bài i rồi lại chuyển sang trạng thái đợi ở đầu cuộn băng. Thời gian quay băng bỏ qua mỗi bài và thời gian phát bài đó được tính là như nhau. Thời gian máy chuyển về đầu cuộn băng là không đáng kể. Tính trung bình, các bài hát trong một ngày được khách hàng lựa chọn với số lần (tần suất) như nhau. Hãy tìm cách ghi các bài trên băng sao cho tổng thời gian quay băng trong mỗi ngày là ít nhất.*

*Dữ liệu vào được ghi trong file văn bản MUSIC.INP.*

*Dòng đầu tiên là số tự nhiên N cho biết số lượng bài hát.*

*Tiếp đến là N số nguyên dương thể hiện dung lượng tính theo phút của mỗi bài. Mỗi đơn vị dữ liệu được ghi cách nhau qua dấu cách.*

Ví dụ

|  |  |
| --- | --- |
| MUSIC.INP  3  7 2 3 | MUSIC.OUT  2 2  3 5  1 12  19 |

Có N = 3 bài hát:

Bài 1 phát trong thời gian 7 phút.

Bài 2 phát trong thời gian 2 phút.

Bài 3 phát trong thời gian 3 phút.

Dữ liệu ra được ghi trong tệp văn bản MUSIC.OUT theo dạng thức sau:

* *N dòng đầu tiên thể hiện trật tự ghi bài hát trên băng: mỗi dòng gồm hai số nguyên dương j và d cách nhau, trong đó j là mã số của bài hát cần ghi, d là thời gian tìm và phát bài đó theo trật tự ghi này.*
* *Dòng thứ N+1 ghi tổng số thời gian quay băng nếu mỗi bài hát được phát một lần trong ngày.*

Với ví dụ trên, kết quả thu được sẽ như sau:

* *Cần ghi lần lượt trên băng các bài theo trật tự : bài 2, bài 3, bài 1;*
* *Để tìm và phát bài 2 cần 2 phút;*
* *Để tìm và phát bài 3 cần 5 phút;*
* *Để tìm và phát bài 1 cần 12 phút;*
* *Tổng thời gian để tìm và phát mỗi bài một lần là: 19 phút.*

## Testing

Có N phần mềm (PM) mã số 1..N cần được kiểm định trên một máy tính, mỗi PM đòi hỏi đúng 1 giờ test. Với mỗi PM ta biết thời hạn phải nộp kết quả test sau khi hoàn thành kiểm định và tiền thưởng thu được nếu nộp kết quả trước hoặc đúng thời điểm quy định. Chỉ có một máy tính trong tay, hãy lập lịch thực hiện test cho N PM trên máy tính sao cho tổng số tiền thưởng thu được là lớn nhất và thời gian hoạt động của máy là nhỏ nhất. Giả thiết rằng máy tính được khởi động vào đầu ca, thời điểm t = 0 và chỉ tắt máy sau khi đã hoàn thành đủ N công việc.

Dữ liệu vào: tệp văn bản TEST.INP:

* Dòng đầu tiên là số N.
* N dòng tiếp theo: mỗi PM cần test được mô tả bằng hai số tự nhiên, số thứ nhất là thời hạn giao nộp kết quả kiểm định, số thứ hai là tiền thưởng. Các số cách nhau qua dấu cách.

Ví dụ

|  |  |
| --- | --- |
| TEST.INP  4  1 15  3 10  5 100  1 27 |  |

Ý nghĩa của dữ liệu vào: Có 4 PM với các thông tin sau:

* *PM thứ nhất phải nộp kết quả test không muộn hơn thời điểm 1 (giờ) với tiền thưởng 15 V (đơn vị tiền tệ);*
* *PM thứ hai phải nộp không muộn hơn thời điểm 3 (giờ) với tiền thưởng 10 V;*
* *PM thứ ba phải nộp không muộn hơn thời điểm 5 (giờ) với tiền thưởng 100 V;*
* *PM thứ tư phải nộp không muộn hơn thời điểm 1 (giờ) với tiền thưởng 27 V.*

*Dữ liệu ra:* tệp văn bản TEST.OUT:

* *N dòng đầu tiên, dòng thứ t ghi một số tự nhiên i cho biết việc thứ i được làm trong giờ t.*
* *Dòng cuối cùng ghi tổng số tiền thu được.*

*Với ví dụ trên, tệp TEST.OUT sẽ như sau:*

|  |
| --- |
| TEST.OUT  4  2  3  1  137 |

*Ý nghĩa của dữ liệu ra*:

* *Giờ thứ 1 test PM 4 và nộp đúng hạn nên được thưởng 27 V;*
* *Giờ thứ 2 test PM 2 và nộp trước hạn nên được thưởng 10 V;*
* *Giờ thứ 3 test PM 3 và nộp trước hạn nên được thưởng 100 V;*
* *Giờ thứ 4 test PM 1 không có thưởng;*

*Tổng tiền thưởng thu được do đã hoàn thành đúng hạn ba PM 4, 2 và 3 là 27 + 10 + 100 = 137 (V).*

## Knapsack

*Có N vật (mặt hàng) mã số 1..N, với mỗi vật ta biết trọng lượng và giá trị của nó. Hãy xác định trọng lượng cần lấy ở một số vật để xếp vào một ba lô có sức chứa tối đa là M kg sao cho giá trị chứa trong ba lô là lớn nhất. Giả thiết là có thể lấy một số kg ở mỗi vật.*

*Dữ liệu vào:* Tệp văn bản KNAPSACK.INP:

* *Dòng đầu tiên: hai giá trị nguyên dương N và M.*
* *N dòng tiếp theo, mỗi dòng chứa hai giá trị nguyên dương d v cho mỗi vật, trong đó d là trọng lượng, v là giá trị tính theo một đơn vị trọng lượng của vật đó (đơn giá). Các số cách nhau qua dấu cách.*

|  |  |  |
| --- | --- | --- |
| KNAPSACK.INP  5 30  8 5  5 4  4 2  3 8  16 6 | *N=5 vật, sức chứa tối đa của ba lô là  M=30 (kg).*  *Vật thứ nhất có trọng lượng 8, đơn giá 5*  *Vật thứ hai có trọng lượng 5, đơn giá 4*  *Vật thứ ba có trọng lượng 4, đơn giá 2,*  *Vật thứ tư có trọng lượng 3, đơn giá 8*  *Vật thứ năm có trọng lượng 16, đơn giá 6*  *Trọng lượng: kg*  *Đơn giá V / kg (V là đơn vị tiền)* | KNAPSACK.OUT    8  3  0  3  16  172 |

*Dữ liệu ra: Tệp văn bản tên* KNAPSACK.OUT:

* *N dòng: dòng thứ i cho biết trọng lượng cần lấy ở vật thứ i.*
* *Dòng cuối cùng ghi tổng giá trị thu được.*

# Bài giải Phương pháp tham

## Music box

Thuật toán

|  |  |  |  |
| --- | --- | --- | --- |
| *Mã số bài hát* | ① | ② | ③ |
| *Thời gian phát* | 7 | 2 | 3 |

Giả sử ta có ba bài hát với thời lượng như sau:

Gọi t(i) là thời gian tìm và phát bài hát thứ i trên băng. Ta có

t(i) = t(1) +...+ t(i-1) + t(i)

trong đó giá trị t(1)+...+t(i-1) là tổng thời gian quay băng bỏ qua i-1 bài hát xếp trên bài i.

Ta xét vài tình huống ghi băng để rút ra kết luận cần thiết.

|  |  |
| --- | --- |
| *Trật tự ghi trên băng*  x y z: t(x) t(y) t(z) | *Thời gian tìm và phát đủ 3 bài*  t(x)+t(y)+t(z) |
| ①②③: 7 2 3 | (7)+(7+2)+(7+2+3) = 28' |
| ①③②: 7 3 2 | (7)+(7+3)+(7+3+2) = 29' |
| ②①③: 2 7 3 | (2)+(2+7)+(2+7+3) = 23' |
| ②③①: 2 3 7 | (2)+(2+3)+(2+3+7) = 19' *(phương án tối ưu)* |
| ③①②: 3 7 2 | (3)+(3+7)+(3+7+2) = 25' |
| ③②①: 3 2 7 | (3)+(3+2)+(3+2+7) = 20' |

Ta thấy rằng bài có thời lượng dài cần để vào cuối dãy. Vậy phương án tối ưu sẽ là ②③①: ghi bài 2 rồi đến bài 3, cuối cùng ghi bài 1. Tổng thời gian theo phương án này là 19 phút.

Để có phương án tối ưu ta chỉ cần ghi băng theo trật tự *tăng dần* của thời lượng. Bài toán được cho với giả thiết băng đủ lớn để ghi được toàn bộ các bài. Dễ dàng sửa chương trình để vận dụng cho trường hợp dung lượng tăng hạn chế trong M phút. Chương trình sắp xếp dữ liệu theo chỉ dẫn.

Chương trình

# Music Box

FN = "MUSIC.INP"

GN = "MUSIC.OUT"

def Go(msg = ' ? '):

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

def Music():

# read input

with open(FN) as f:

b = list(map(int,f.read(). split()))

n = b[0]

print(' Số bài hát: ', n)

# thời lượng

a = b[1:]

print(' Thời lượng mỗi bài hát a: ', a)

id = list(range(n))

id.sort(key = lambda i: a[i]) # sắp tăng thời lượng thao id

t = [0]\*n # thời lượng phát mỗi bài

t[0] = a[id[0]]

for i in range(1,n): t[i] = t[i-1] + a[id[i]]

print(t)

# write result

with open(GN, 'w') as g:

for i in range(n):

g.write(str(id[i]+1) + " " + str(t[i]) + "\n")

g.write(str(sum(t)) + "\n")

# APPLICATION

Music()

print(' T h e E n d')

MUSIC.OUT

2 2

3 5

1 12

19

Độ phức tạp

n log n chi phí cho viẹc sắp tăng n bài hát theo thời lượng.

## Testing

Thuật toán

Ta ưu tiên cho những phần mềm (PM) có tiền thưởng cao, do đó ta sắp các PM giảm dần theo tiền thưởng. Với mỗi PM v ta đã biết thời hạn giao nộp kết quả test PM đó là i = t[v]. Ta xét trục thời gian h. Nếu giờ i trên trục đã bận do PM khác chiếm chỗ thì ta tìm từ thời điểm i trở về trước một thời điểm j có thể thực hiện được PM v đó. Nếu tìm được một thời điểm j như vậy, ta hiểu là PM này sẽ có thưởng. Ta đánh dấu bằng mã số của nó trên trục thời gian h, h[j] = v. Sau khi đã duyệt xong các PM, có thể trên trục thời gian còn những thời điểm rỗi, ta dồn các PM đã xếp về phía trước nhằm thu được một lịch làm việc trù mật, tức là không có giờ trống. Cuối cùng ta xếp tiếp những PM trước đó đã xét nhưng không xếp được. Đây là những PM phải làm nhưng không thể nộp đúng hạn nên sẽ không có tiền thưởng. Với ví dụ đã cho, N = 4,thời hạn giao nộp t = (1, 3, 5, 1) và tiền thưởng a = (15, 10, 100, 27) ta tính toán như sau*:*

* Khởi trị: trục thời gian với 5 thời điểm ứng với Tmax = 5 là thời điểm muôn nhất phải nộp kết quả, Tmax = max {*thời hạn giao nộp* }

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *i* |  | *1* | *2* | *3* | *4* | *5* |
| *h* |  | -1 | -1 | -1 | -1 | -1 |
| *thưởng* |  |  |  |  |  |  |
| *Khởi trị* | | | | | | |

* Chọn PM 3 có tiền thưởng lớn nhất là 100. Xếp PM 3 với thời hạn *t*[3] = 5 vào *h*: *h*[5] = 3.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *i* | *0* | *1* | *2* | *3* | *4* | *5* |
| *h* | -1 | -1 | -1 | -1 | -1 | 3 |
| *thưởng* |  |  |  |  |  | 100 |
| *Xếp việc 3* | | | | | | |

* Chọn tiếp PM 4 có tiền thưởng 27. Xếp PM 4 với thời hạn *t*[4] = 1 vào *h*: *h*[1] = 4.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *i* |  | *1* | *2* | *3* | *4* | *5* |
| *h* |  | 4 | -1 | -1 | -1 | 3 |
| *thưởng* |  | 27 |  |  |  | 100 |
| *Xếp việc 4* | | | | | | |

* Chọn tiếp PM 1 có tiền thưởng 15. Xếp PM 1 với thời hạn *t*[1] = 1 vào *h*: Không xếp được vì từ thời điểm 1 trở về trước trục thời gian *h* đã kín.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *i* |  | *1* | *2* | *3* | *4* | *5* |
| *h* |  | 4 | -1 | -1 | -1 | 3 |
| *thưởng* |  | 27 |  |  |  | 100 |
| *Xét việc 1 ?* | | | | | | |

* Chọn nốt PM 2 có tiền thưởng 10. Xếp PM 2 với thời hạn *t*[2] = 3 vào *h*: *h*[3] = 2.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *i* |  | *1* | *2* | *3* | *4* | *5* |
| *h* |  | 4 | -1 | 2 | -1 | 3 |
| *thưởng* |  | 27 |  | 10 |  | 100 |
| *Xếp việc 2* | | | | | | |

* Dồn các PM trên trục thời gian *h*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *i* |  | *1* | *2* | *3* | *4* | *5* |
| *h* |  | 4 | 2 | 3 | -1 | -1 |
|  |  |  |  |  |  |  |
| *Dồn PM* | | | | | | |

* Xếp nốt các PM phải làm mà không có thưởng, ta thu được *h* = (4, 2, 3, 1).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *i* |  | *1* | *2* | *3* | *4* | *5* |
| *h* |  | 4 | 2 | 3 | 1 | -1 |
|  |  |  |  |  |  |  |
| *Xếp nốt các PM còn lại* | | | | | | |

* Ca làm việc kéo dài đúng *N* = 4 giờ.
* Tổng tiền thưởng: 137

Vì các chỉ số mảng được tính từ 0 nên lúc đầu ta mã số các PM từ 0, khi ghi kết quả ta sẽ tăng chỉ số thêm 1.

Chương trình

# Testing

FN, GN = "TEST.INP", "TEST.OUT"

def Go():

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

def Init():

global id

# sắp giảm theo tiền thưởng

id = list(range(n))

id.sort(key = lambda i: thuong[i], reverse = True)

print(' Sau khi sắp giảm theo tiền thưởng')

print(' CHÚ Ý: các chỉ số bị dịch -1')

print(' PM:Thưởng:Hạn nộp')

for i in range(n):

j = id[i]

print(' ', str(j)+':'+str(thuong[j])+':'+str(nop[j]))

def ReadInput():

global n, nop, thuong

with open(FN) as f:

n = int(f.readline())

print(n, ' PM ')

nop = [0]\*n # thừi hạn

thuong = [0]\*n # tiền thưởng

for i in range(n):

nop[i], thuong[i] = map(int, f.readline().split())

print(' Hạn nộp:', nop); print(' Thưởng:', thuong)

# Uu tien PM thuong cao

def Xep():

global h, tongThuong

tmax = max(nop)

print(' Tmax:', tmax)

h = [-1]\*(tmax+1) # trục thời gian

chuaXep = [] # những PM chưa xếp được

tongThuong = 0 # tổng tiền thưởng

for i in range(n):

v = id[i]

print(' Xét PM No ', v, ' Thưởng', thuong[v], end = '')

print(' Hạn nộp', nop[v])

j = TimChoXep(v)

if j > 0:

h[j] = v

tongThuong += thuong[v]

print(' . Xếp được tại thời điểm ', j)

else: print(' . Không xếp được'); chuaXep.append(v)

print(' Tổng tiền thưởng', tongThuong)

print(' Các PM chưa xếp', chuaXep)

print(' Dồn các PM đã xếp trên trục h')

m = 0 # số lượng PM sau khi dồn

for i in range(1, tmax+1):

if h[i] != -1:

h[m] = h[i]

h[i] = -1

m += 1

print(h)

print(' Xếp nốt các PM còn lại (không thưởng):', chuaXep)

for v in chuaXep: h[m] = v; m += 1

print(h)

def TimChoXep(v):

for i in range(nop[v], 0, -1):

if h[i] == -1: return i

return 0

def WriteResult():

with open(GN,'w') as g:

for v in h:

if v < 0: break

g.write(str(v+1)+"\n")

g.write(str(tongThuong)+"\n")

print('Đã ghi file:')

with open(GN) as f: print(f.read().strip())

def Run():

ReadInput()

Init()

Xep()

WriteResult()

# APPLICATION

Run()

print('\n T h e E n d')

Output

4 PM

Hạn nộp: [1, 3, 5, 1]

Thưởng: [15, 10, 100, 27]

Sau khi sắp giảm theo tiền thưởng

CHÚ Ý: các chỉ số bị dịch -1

PM:Thưởng:Hạn nộp

2:100:5

3:27:1

0:15:1

1:10:3

Tmax: 5

Xét PM No 2 Thưởng 100 Hạn nộp 5

. Xếp được tại thời điểm 5

Xét PM No 3 Thưởng 27 Hạn nộp 1

. Xếp được tại thời điểm 1

Xét PM No 0 Thưởng 15 Hạn nộp 1

. Không xếp được

Xét PM No 1 Thưởng 10 Hạn nộp 3

. Xếp được tại thời điểm 3

Tổng tiền thưởng 137

Các PM chưa xếp [0]

Dồn các PM đã xếp trên trục h

[3, 1, 2, -1, -1, -1]

Xếp nốt các PM còn lại (không thưởng): [0]

[3, 1, 2, 0, -1, -1]

Đã ghi file:

4

2

3

1

137

T h e E n d

Độ phức tạp

Pha sắp giảm *n* PM theo tiền thưởng có độ phức tạp n log n, pha duyệt n PM cần xác định n thời điểm theo trục thời gian h đòi hỏi chi phí . Vậy độ phức tạp của bài là O().