

SỞ GIÁO DỤC VÀ ĐÀO TẠO THANH HOÁ
TRƯỜNG THPT HÀM RỒNG

SÁNG KIẾN KINH NGHIỆM

MỘT SỐ BÀI TOÁN QUY HOẠCH ĐỘNG TRONG
BỒI DƯỠNG HSG TIN HỌC 11

Người thực hiện: Nguyễn Thị Mai Hương
Chức vụ: Giáo viên
SKKN thuộc lĩnh vực (môn): Tin học

THANH HOÁ NĂM 2019

MỤC LỤC

1. MỞ ĐẦU	1
1.1. Lý do chọn đề tài.....	1
1.2. Mục đích nghiên cứu.....	1
1.3. Đối tượng và phạm vi nghiên cứu.....	1
1.4. Phương pháp nghiên cứu.....	1
2. NỘI DUNG SÁNG KIẾN KINH NGHIỆM	2
2.1. PHƯƠNG PHÁP QUY HOẠCH ĐỘNG	2
2.2. THỰC TRẠNG TRƯỚC KHI ÁP DỤNG SÁNG KIẾN KINH NGHIỆM.....	2
2.3. MỘT SỐ BÀI TOÁN ÁP DỤNG	3
Bài 1: Bội số chung nhỏ nhất	3
Bài 2: Dãy con đối xứng dài nhất	5
Bài 3: Thả bóng	7
2.4. HIỆU QUẢ CỦA SÁNG KIẾN KINH NGHIỆM ĐỐI VỚI HOẠT ĐỘNG GIÁO DỤC, VỚI BẢN THÂN, ĐỒNG NGHIỆP VÀ NHÀ TRƯỜNG	10
3. KẾT LUẬN, KIẾN NGHỊ.....	11

TÊN ĐỀ TÀI:
MỘT SỐ BÀI TOÁN QUY HOẠCH ĐỘNG TRONG BỒI DƯỠNG
HSG TIN HỌC 11

1. MỞ ĐẦU

1.1. Lý do chọn đề tài

Để giải các bài toán trong Tin học, việc xác định Thuật toán để giải bài toán là công việc đặc biệt quan trọng. Tuy nhiên, đối với học sinh các trường THPT việc thiết kế thuật toán để giải quyết tối ưu bài toán gặp nhiều khó khăn.

Trong các đề thi HSG Tin học cấp Tỉnh, có một số bài toán cần sử dụng những kĩ thuật lập trình nâng cao. Mặc dù, cũng có những cách khác để giải bài toán đó nhưng vì các cuộc thi đều có giới hạn về thời gian, cũng như bộ nhớ của chương trình, nên một thuật toán hiệu quả là cực kỳ cần thiết. Và trong những trường hợp như vậy, quy hoạch động là một lựa chọn tốt. Do vậy, tôi đã chọn đề tài “Một số bài toán quy hoạch động trong bồi dưỡng học sinh giỏi Tin học lớp 11”.

1.2. Mục đích nghiên cứu

- Nhằm mục đích đóng góp thêm vào ngân hàng bài tập giúp hỗ trợ cho công tác bồi dưỡng học sinh giỏi môn Tin học được tốt hơn.

1.3. Đối tượng và phạm vi nghiên cứu

- Môn Tin học lớp 11 ở trường THPT;
- Học sinh khối 11 trường THPT Hàm Rồng;

1.4. Phương pháp nghiên cứu

- Phân tích, tổng hợp, khảo sát.
- Đánh giá so sánh kết quả của học sinh;

2. NỘI DUNG SÁNG KIẾN KINH NGHIỆM

2.1. PHƯƠNG PHÁP QUY HOẠCH ĐỘNG

Trong chiến lược chia để trị, người ta phân bài toán cần giải thành các bài toán con. Các bài toán con lại tiếp tục phân thành các bài toán con nhỏ hơn, cứ thế tiếp tục cho tới khi ta nhận được các bài toán con có thể giải được dễ dàng. Tuy nhiên, trong quá trình phân chia như vậy, có thể ta sẽ gặp rất nhiều lần cùng một bài toán con. Tư tưởng cơ bản của phương pháp quy hoạch động là sử dụng một bảng để lưu giữ lời giải các bài toán con đã được giải. Khi giải một bài toán con cần đến nghiệm của bài toán con cỡ nhỏ hơn, ta chỉ cần lấy lời giải ở trong bảng mà không cần phải giải lại. Chính vì thế mà các thuật toán được thiết kế bằng quy hoạch động sẽ rất hiệu quả.

Khi nào thì chúng ta cần đến quy hoạch động? Không có một công thức nào cho các bài toán như vậy. Tuy nhiên, có một số tính chất của bài toán mà ta có thể nghĩ đến quy hoạch động. Dưới đây là hai tính chất nổi bật nhất trong số chúng:

- Bài toán có các bài toán con gối nhau.
- Bài toán có cấu trúc con tối ưu.

Thường thì một bài toán có đủ cả hai tính chất này, chúng ta có thể dùng quy hoạch động được.

Để giải quyết một bài toán bằng quy hoạch động, chúng ta cần tiến hành những công việc sau:

- Tìm nghiệm của các bài toán con nhỏ nhất;
- Tìm ra công thức (hoặc quy tắc) xây dựng nghiệm của bài toán con thông qua nghiệm của các bài toán con cỡ nhỏ hơn;
- Tạo ra một bảng lưu trữ các nghiệm của bài toán con. Sau đó tính nghiệm của các bài toán con theo công thức đã tìm và lưu vào bảng;
- Từ các bài toán con đã giải để tìm nghiệm của bài toán.

2.2. THỰC TRẠNG TRƯỚC KHI ÁP DỤNG SÁNG KIẾN KINH NGHIỆM

Trong quá trình bồi dưỡng học sinh giỏi Tin học 11 có một số bài tập lập trình nếu chỉ áp dụng những thuật toán cơ bản thì thời gian thực hiện chương trình không đảm bảo. Đây cũng là khó khăn lớn đối với học sinh vì các em chưa được làm quen nhiều với các bài tập thiết kế thuật toán quy hoạch động. Vì vậy việc tập hợp những bài toán quy hoạch động thành một hệ thống bài tập để học sinh tham khảo là rất cần thiết.

2.3. MỘT SỐ BÀI TOÁN ÁP DỤNG

Bài 1: Bộ số chung nhỏ nhất

Cho trước số tự nhiên N ($1 < N < 100$)

Xét tất cả các phân tích N thành tổng các số tự nhiên: $N = a_1 + a_2 + \dots + a_k$ (1)

Tìm giá trị lớn nhất của BSCNN (a_1, a_2, \dots, a_k) trên tập các bộ số (a_1, a_2, \dots, a_k) thỏa mãn đẳng thức (1).

Dữ liệu vào: từ file văn bản BSCNN.INP là 1 số N

Dữ liệu ra: ghi ra file văn bản BSCNN.OUT, bao gồm 2 dòng, dòng thứ nhất ghi giá trị Max tìm được. Dòng thứ hai ghi n bộ số a_1, a_2, \dots, a_k tương ứng, các số cách nhau bằng dấu cách.

Ví dụ:

BSCNN.INP	BSCNN.OUT
22	420 3 3 4 5 7

a) Phân tích thuật toán:

Áp dụng thuật toán quy hoạch động:

Gọi $a[i,j]$ là giá trị lớn nhất của BSCNN khi phân tích i thành tổng của j số tự nhiên.

$B[i,j]$ là giá trị của số thứ j trong cách phân tích i thành tổng của j số tự nhiên để $a[i,j]$ đạt max.

Khởi tạo: $a[i,1]:=i$; $b[i,1]:=i$ ($1 \leq j \leq i \leq N$)

Với mọi j ($2 \leq j \leq N$) ta xét các khả năng phân tích i ($j \leq i \leq N$) để BSCNN của các cách phân tích đó đạt max.

Ta có: $a[i,j]:= \max \{BSCNN(a[i-t,j-1],t)\}$ với $1 \leq t \leq i-j+1$; $b[i,j]:=t$;

Giá trị lớn nhất trong các cách phân tích N bằng $\max(a[n,i])$ với $1 \leq j \leq N$

b) Chương trình tham khảo:

```
program BSCNN;
const  fi='BSCNN.INP';
       fo='BSCNN.OUT';
       maxn=100;
var  a: array[1..maxn,1..maxn] of longint;
     b: array[1..maxn,1..maxn] of byte;
     n: integer;
     f: text;
{-----}
procedure init;
begin
assign(f, fi); reset(f);
readln(f, n);
```

```

close(f);
end;
{-----}
function bscnn(x,y: longint): longint;
var tg,i,j: longint;
begin
  i:=x;
  j:=y;
  while i mod j<>0 do
    begin
      tg:= j;
      j:= i mod j;
      i:=tg;
    end;
  bscnn:= (x*y)div j;
end;
{-----}
procedure solution;
var i,j,t: integer;
    tg: longint;
begin
  for i:=1 to n do
    begin
      a[i,1]:=i;b[i,1]:=i;
    end;
  for j:=2 to n-1 do
    for i:=j+1 to n do
      begin
        a[i,j]:=0;
        for t:=1 to i-j+1 do
          begin
            tg:=bscnn(a[i-t,j-1],t);
            if a[i,j]<tg then
              begin
                a[i,j]:=tg;
                b[i,j]:=t;
              end;
          end;
        end;
      end;
    end;
end;
{-----}
procedure result;
var i,j: integer;
begin
  assign(f,fo);
  rewrite(f);
  j:=n;
  for i:=1 to n-1 do
    if a[n,n]< a[n,i] then
      begin
        a[n,n]:=a[n,i];
        j:=i;
      end;
  end;

```

```

writeln(f, a[n, n]);
i:=n;
while j>0 do
  begin
    write(f, b[i, j], ' ');
    i:=i-b[i, j];
    j:=j-1;
  end;
close(f);
end;
{-----}
BEGIN
  init;
  solution;
  result;
END.

```

Bài 2: Dãy con đối xứng dài nhất

Dãy số có A_1, A_2, \dots, A_n được gọi là đối xứng nếu các cặp số ở các vị trí i và $N-i+1$ bằng nhau (với $i=1..N$). Cho trước một dãy số có N phần tử, mỗi phần tử là số nguyên. Hãy tìm cách loại bỏ một số phần tử trong dãy để dãy thu được tạo thành một dãy đối xứng dài nhất.

Dữ liệu vào: tệp văn bản DAYDX.INP có cấu trúc như sau:

- + Dòng 1 chứa số nguyên N ($2 \leq N \leq 5000$);
- + Dòng thứ 2 ghi N số nguyên là các số hạng trong dãy có giá trị tuyệt đối ≤ 1000 , mỗi số cách nhau một dấu cách.

Dữ liệu ra: tệp văn bản DAYDX.OUT với cấu trúc như sau:

- + Dòng đầu ghi số nguyên M là số các số hạng của dãy đối xứng tìm được;
- + Dòng thứ 2 ghi M số hạng của dãy tìm được, mỗi số cách nhau một dấu cách.

Ví dụ:

DAYDX.INP	DAYDX.OUT
13	7
1 3 2 3 1 5 2 3 4 1 4 3 2	2 3 4 1 4 3 2

a) Phân tích thuật toán:

Áp dụng thuật toán quy hoạch động:

Gọi $L[i, j]$ ($i=1..N, j=1..N$) là độ dài lớn nhất của dãy tìm được khi chọn các phần tử của A từ phần tử thứ i đến j . Ta cần tìm $L[1, N]$

*) Công thức quy hoạch động:

$L[i, j]=0$ với mọi $i > j$;

$L[i, j]=1$ với $i=j$;

Nếu $A[i]=A[j]$ thì $L[i,j]=L[i+1,j-1]+2$, ngược lại $L[i,j]=\max(L[i,j-1], L[i+1,j])$;

Sử dụng mảng hai chiều $L[1..N, 1..N]$ ($2 \leq N \leq 5000$)

Điền các phần tử của mảng nằm trên đường chéo chính $k=-1$ đến $1-N$ với điều kiện: Nếu $A[i]=A[j]$ thì $L[i,j]=L[i+1,j-1]+2$, ngược lại $L[i,j]=\max(L[i,j-1], L[i+1,j])$;

Kết quả của bài toán là $L[1,N]$.

Dựa vào mảng L lần ngược từ giá trị $L[1,N]$ để lấy dãy các phần tử tìm được.

b) Chương trình tham khảo:

```
const
  fi='DAYDX.INP';
  fo='DAYDX.OUT';
  nmax=5000;
var
  a:array[1..nmax] of integer;
  check: array[1..nmax] of boolean;
  l:array[1..nmax,1..nmax] of integer;
  n:integer;
{-----}
procedure inputdata;
var f:text; i:integer;
begin
  assign(f,fi);
  reset(f);
  readln(f,n);
  for i:=1 to n do read(f,a[i]);
  close(f);
  fillchar(check,sizeof(check),false);
  fillchar(l,sizeof(l),0);
  for i:=1 to n do l[i,i]:=1;
end;
{-----}
function max(x,y:integer):integer;
begin
  if x>y then exit(x) else exit(y);
end;
{-----}
procedure qhd;
var i,j,k:integer;
begin
  k:=0;
  repeat
    dec(k);
    for i:=1 to n do
      begin
        j:=i-k;
        if a[i] = a[j] then l[i,j]:=l[i+1,j-1]+2
```



```

else l[i,j]:=max(l[i,j-1],l[i+1,j])
end;
until k=1-n;
end;
{-----}
procedure trace;
var i,j:integer;
begin
i:=1;j:=n;
repeat
while l[i,j]=l[i+1,j] do inc(i);
while l[i,j]=l[i,j-1] do dec(j);
check[i]:=true; check[j]:=true;
inc(i);dec(j);
until i>j
end;
{-----}
procedure outputdata;
var f:text; i,j:integer;
begin
assign(f,fo);
rewrite(f);
writeln(f,l[1,n]);
for i:=1 to n do
if check[i] then write(f,a[i],' ');
close(f);
end;
{-----}
BEGIN
inputdata;
qhd;
trace;
outputdata;
END.

```

Bài 3: Thả bóng

Kỉ niệm ngày 26/3, Đoàn trường có tổ chức trò chơi thả bóng. Người chơi đứng lên cao và thả quả bóng xuống một các bảng bậc thang chữ nhật. Bảng này được chia thành $n \times m$ hình chữ nhật nhỏ, mỗi hình chữ nhật nhỏ có gắn một bậc thang và trên đó có ghi một số nguyên hoặc đặt một cái đinh làm bằng tấm bìa cứng. Người chơi đứng trên thả quả bóng xuống, quả bóng chỉ được lăn theo quy tắc: từ ô (i,j) chỉ được lăn xuống các ô $(i+1,j-1)$, $(i+1,j)$, $(i+1,j+1)$, nếu đi sai quy tắc thì sẽ bị mất quyền chơi còn nếu đi đúng thì mỗi ô mà quả bóng lăn qua sẽ được cộng vào quỹ điểm của người chơi bằng số nguyên được ghi trên ô, còn nếu lăn vào ô có đinh thì số điểm của người chơi sẽ bị trừ đi một nửa hoặc gần một nửa (phần nguyên của số điểm chia 2). Sau khi hoàn thành trò chơi thì số điểm sẽ được

quy ra quà cho người chơi. Để khuyến khích người chơi, ban tổ chức sẽ tặng người chơi một số điểm nào đó.

Yêu cầu: Hãy cho người chơi biết có thể nhận được số điểm ít nhất và nhiều nhất mà họ có thể nhận được.

Dữ liệu vào: từ file THABONG.INP

- Dòng đầu chứa 2 số nguyên n và m ($1 \leq n, m \leq 1000$) là giá trị của các phần tử của dãy A .

- Dòng thứ hai chứa số nguyên b là số điểm mà ban tổ chức tặng người chơi ($b < 30000$).

- Dòng thứ i trong số n dòng tiếp theo chứa m số nguyên không âm là m số của hàng i ; nếu ô ghi số 0 có nghĩa là ô đó có đỉnh còn ngược lại là các số trên ô.

Dữ liệu ra: ghi ra file THABONG.OUT

- Dòng đầu ghi số điểm ít nhất mà người chơi đạt được;

- Dòng hai ghi số điểm nhiều nhất mà người chơi đạt được.

Ví dụ:

THABONG.INP	THABONG.OUT
5 5	18
100	148
1 7 5 10 2	
6 8 4 1 9	
9 6 0 7 14	
10 18 3 2 0	
5 0 6 11 10	

a) Phân tích thuật toán: sử dụng phương pháp quy hoạch động

- Tìm giá trị lớn nhất:

+ Gọi $L[i,j]$ là giá trị lớn nhất từ ô ở hàng tới ô (i,j)

+ Khi đó ta có: $L[i,j] = \max(L[i-1,j-1], L[i-1,j], L[i-1,j+1]) + \text{giá trị tại ô } (i,j)$

($i=2, 3, \dots, n; j=1, 2, \dots, m$);

Ban đầu ta đặt: $L[1,j] = \text{giá trị tại ô } (1,j) + p$ với $j=1, 2, \dots, m$

+ Tìm giá trị max tại ô ở hàng n của bảng L

- Tìm giá trị nhỏ nhất:

+ Gọi $T[i,j]$ là giá trị nhỏ nhất từ ô ở hàng 1 tới ô (i,j)

+ Khi đó ta có: $T[i,j] = \min(T[i-1,j-1], T[i-1,j], T[i-1,j+1]) + \text{giá trị tại ô } (i,j)$

($i=2, 3, \dots, n; j=1, 2, \dots, m$); nếu giá trị của ô $(i,j)=0$ thì $T[i,j] = T[i,j] - T[i,j] \text{ div } 2$

Ban đầu ta đặt: $T[1,j] = \text{giá trị tại ô } (1,j) + p$ với $j=1, 2, \dots, m$; nếu giá trị của ô $(i,j)=0$ thì $T[i,j] = p \text{ div } 2$

+ Ta tìm giá trị min tại ô ở hàng n của bảng T.

b) Chương trình tham khảo:

```
Program ThaBong;
const
  fi='THABONG.INP';
  fo='THABONG.OUT';
  nmax=1000;
var
  a:array[1..nmax,1..nmax] of integer;
  L:array[1..nmax,1..nmax] of longint;
  n,m,p:integer;
  Hmax,Hmin:longint;
{-----}
procedure inputdata;
var f:text; i,j:integer;
begin
  assign(f,fi);
  reset(f);
  readln(f,n,m);
  readln(f,p);
  for i:=1 to n do
    for j:=1 to m do read(f,a[i,j]);
  close(f);
end;
{-----}
function max(a,b,c:integer):integer;
var tg:integer;
begin
  tg:=a;
  if tg<b then tg:=b;
  if tg<c then tg:=c;
  max:=tg;
end;
{-----}
function min(a,b,c:integer):integer;
var tg:integer;
begin
  tg:=a;
  if tg>b then tg:=b;
  if tg>c then tg:=c;
  min:=tg;
end;
{-----}
function cottr(c:integer):integer;
begin
  if c<=1 then cottr:=1 else cottr:=c-1;
end;
{-----}
function cotph(c:integer):integer;
begin
  if c>=1 then cotph:=m else cotph:=c+1;
end;
{-----}
```

```

procedure process;
var i,j:integer;
begin
    for j:=1 to m do L[1,j]:=p+a[1,j];
    for i:=2 to n do
        for j:=1 to m do
            L[i,j]:=max(L[i-1,cottr(j)],L[i-1,j],L[i-1,cotph(j)])+a[i,j];
        Hmax:=L[n,1];
        for j:=1 to m do
            if a[1,j]=0 then L[1,j]:=p-p div 2 else
L[1,j]:=p+a[1,j];
        for i:=2 to n do
            for j:=1 to m do
                begin
                    L[i,j]:=min(L[i-1,cottr(j)],L[i-1,j],L[i-1,cotph(j)])+a[i,j];
                    if a[i,j]=0 then L[i,j]:=L[i,j]-L[i,j] div 2;
                end;
            Hmin:=L[n,1];
            for j:=2 to m do
                if Hmin>L[n,j] then Hmin:=L[n,j];
        end;
    {-----}
procedure outputdata;
var f:text;
begin
    assign(f,fo);
    rewrite(f);
    writeln(f,Hmin);
    writeln(f,Hmax);
    close(f);
end;
{-----}
BEGIN
    inputdata;
    process;
    outputdata;
END.

```

2.4. HIỆU QUẢ CỦA SÁNG KIẾN KINH NGHIỆM ĐỐI VỚI HOẠT ĐỘNG GIÁO DỤC, VỚI BẢN THÂN, ĐỒNG NGHIỆP VÀ NHÀ TRƯỜNG

Với sáng kiến kinh nghiệm này, theo ý kiến chủ quan của bản thân tôi nhận thấy đã đóng góp được một phần nhỏ vào công tác bồi dưỡng học sinh giỏi môn tin học tại đơn vị công tác của mình- Trường THPT Hàm Rồng. Giúp trang bị thêm kiến thức cho các em học sinh trong đội tuyển Tin để các em có được động lực, sự tự tin để học tốt hơn. Giúp các đồng nghiệp có thêm tài liệu tham khảo, cùng nhau tự học, tự bồi dưỡng.

3. KẾT LUẬN, KIẾN NGHỊ

Hệ thống lý thuyết và một số bài tập về quy hoạch động được đưa ra để học sinh có thể luyện tập và nâng cao kỹ năng lập trình của mình phải tương đối lớn và được lựa chọn thật kỹ càng, thêm nữa các bài toán phải được phân loại và tập hợp thành những module cụ thể, điều này sẽ gây hứng thú hơn cho học sinh trong quá trình nghiên cứu và vận dụng. Tuy nhiên, do thời gian có hạn nên trong đề tài chỉ mới đưa ra một số ít các bài toán về quy hoạch động.

Mặc dù bản thân đã cố gắng nhiều, song trong đề tài còn nhiều khiếm khuyết, tôi rất mong nhận được sự đóng góp ý kiến của các đồng nghiệp để hoàn thiện hơn và đạt hiệu quả trong giảng dạy và học tập của học sinh.

Tôi xin cam đoan đây là SKKN của mình viết, không sao chép nội dung của người khác./.

XÁC NHẬN CỦA HIỆU TRƯỞNG *Thanh Hoá, ngày 05 tháng 05 năm 2019*

Tôi xin cam đoan đây là SKKN của bản thân viết, không sao chép nội dung của người khác.

Người viết đề tài

Nguyễn Thị Mai Hương

TÀI LIỆU THAM KHẢO

1. SGK Tin học 11- NXB Giáo Dục Việt Nam
2. SBT Tin học 11- NXB Giáo Dục Việt Nam
3. Tài liệu chuyên Tin học- Quyển 1- - NXB Giáo Dục Việt Nam
4. Internet

**DANH MỤC CÁC ĐỀ TÀI SÁNG KIẾN KINH NGHIỆM
ĐÃ ĐƯỢC ĐÁNH GIÁ**

Họ và tên: Nguyễn Thị Mai Hương
Chức vụ công tác: Giáo viên

STT	TÊN ĐỀ TÀI	Cấp đánh giá xếp loại	NĂM HỌC	XẾP LOẠI
1	Một số bài toán bồi dưỡng HSG Tin học 11 về số nguyên tố.	Cấp Sở GD&ĐT	2014 - 2015	C
2	Một số bài toán bồi dưỡng HSG Tin học 11 về xử lí số nguyên lớn.	Cấp Sở GD&ĐT	2016 - 2017	C