**BÁO CÁO BÀI TẬP MÔN HỌC**

**PHÂN TÍCH VÀ THIẾT KẾ GIẢI THUẬT**

***Họ Tên: Lê Thúy Quỳnh***

***Lớp: TH12B***

***Mã đề: 13***

***Đề bài: Đề 13:***

Thiết kế thuật toán tìm dãy con liên tiếp có tổng lớn nhất theo phương pháp qui hoạch động với các công việc sau:

1. Nêu bài toán;

2. Mô tả chi tiết thuật toán;

3. Tự xác định 5 bộ dữ liệu (với phần tử N>=10), với mỗi bộ dữ liệu hãy thực hiện từng bước thuật toán đã mô tả ở mục 2 và ghi ra kết quả mỗi bước;

4. Viết chương trình sử dụng C, C++;

5. Đánh giá độ phức tạp thuật toán theo lý thuyết và bằng thực nghiệm, so sánh kết quả;

6. Viết báo cáo (trình bày các nội dung từ 1-5).

1. **Nêu bài toán:**

***Đầu vào:*** Mảng N số A[1….N]

***Đầu ra:*** Hãy tìm dãy con các phần tử liên tiếp của A có tổng lớn nhất.

***Phương pháp:*** Theo phương pháp Quy hoạch động.

**II. Mô tả chi tiết thuật toán:**

+Thuật toán: Tìm dãy con liên tiếp có tổng lớn nhất theo phương pháp Quy hoạch động.

+Thuật toán Quy hoạch động là một kỹ thuật thiết kế thuật toán theo kiểu chia bài toán lớn thành các bài toán con, sử dụng lời giải của các bài toán con để tìm lời giải cho bài toán ban đầu. Hay nói một cách khác, lược đồ chung của thuật toán là: phân rã (chia bài toán ban đầu thành bài toán con nhỏ nhất có thể) => giải bài toán con và ghi kết quả => tổng hợp lời giải.

***+Chi tiết thuật toán:***

***1.Phân rã:***

-Gọi MaxS[i] là tổng lớn nhất của dãy con liên tiếp có i phần tử a[1]…a[i].

-Khi đó MaxS[N] là giá trị lớn nhất của dãy con liên tiếp cần tìm.

***2. Bài toán cơ sở:***

-Với i=1 ta có: MaxS[i] = a[i]

***3. Tổng hợp:***

-Giả sử i > 1 và MaxS[k] là đã biết với k = 1,….., i-1

Ta cần tính MaxS[i] là tổng của dãy con liên tiếp lớn nhất của dãy a[1],…., a[i-1], a[i].

* Các dãy con liên tiếp của dãy này có thể là 1 trong 2 trường hợp:

TH1: Các dãy con liên tiếp có chứa a[i].

TH2: Các dãy con liên tiếp không chứa a[i].

-Gọi MaxE[i] là tổng lớn nhất của các dãy con liên tiếp của dãy a[1]…..a[i] có chứa a[i].

-Tổng lớn nhất của các dãy con liên tiếp của dãy a[1]…a[i] không chứa a[i] chính là tổng lớn nhất của các dãy con a[1]…a[i-1], nghĩa là MaxS[i-1].

=> MaxS[i] = max{MaxS[i-1], MaxE[i]}

***+Cài đặt:***

Với: s – chỉ số đầu

e – chỉ số cuối

s1 – chỉ số đầu tạm

**Procedu subMax**

{

MaxS=a[1]; MaxE=a[1];

s=1; e=1; s1=1;

for i=2 to n do

{

If MaxE>0 then MaxE=MaxE+a[i];

else {MaxE=a[i]; s1=i;}

if (MaxE>MaxS) then {

MaxS=MaxE;

e=i; s=s1;

}

}

}

**III.Ví dụ cụ thể:**

***VD1: Đầu vào: Mảng 10 số A[1….10] = 1, 2, 18, -14, 3, 7, -18, 5, 9, 12***

***Đầu ra: Hãy tìm dãy con các phần tử liên tiếp của A có tổng lớn nhất.***

***Phương pháp: Theo phương pháp Quy hoạch động.***

***Lời giải:***

B1: Phân rã:

-Gọi MaxS[i] là tổng lớn nhất của dãy con liên tiếp có i phần tử a[1]…a[i].

-Khi đó MaxS[10] là giá trị lớn nhất của dãy con liên tiếp cần tìm.

B2: Bài toán cơ sở:

-Với i=1 ta có: MaxS[1] = a[1] = 1

B3: Tổng hợp:

Với: s: chỉ số đầu, e: chỉ số cuối, s1: chỉ số đầu tạm và:

Nếu MaxE[i-1] > 0 thì MaxE[i] = MaxE[i-1] + a[i]

Nếu MaxE[i-1] < 0 thì MaxE[i] = a[i]

MaxS[i] = max{MaxS[i-1], MaxE[i]}

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| i | a[i] | MaxE[i] | s1 | MaxS[i] | s | e | Dãy con tổng lớn nhất |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 2 | 3 | 1 | 3 | 1 | 2 | 1, 2 |
| 3 | 18 | 21 | 1 | 21 | 1 | 3 | 1, 2, 18 |
| 4 | -14 | 7 | 1 | 21 | 1 | 3 | 1, 2, 18 |
| 5 | 3 | 10 | 1 | 21 | 1 | 3 | 1, 2, 18 |
| 6 | 7 | 17 | 1 | 21 | 1 | 3 | 1, 2, 18 |
| 7 | -18 | -1 | 1 | 21 | 1 | 3 | 1, 2, 18 |
| 8 | 5 | 5 | 8 | 21 | 1 | 3 | 1, 2, 18 |
| 9 | 9 | 14 | 8 | 21 | 1 | 3 | 1, 2, 18 |
| 10 | 12 | 26 | 8 | 26 | 8 | 10 | 5, 9, 12 |

* Dãy con cần tìm là: A[8]-A[10]: 5, 9, 12

***VD2: Đầu vào: Mảng 12 số A[1….12] = 3, 5, -9, 10, 8, 6, 4, -15, -6, -4, 1, 12***

***Đầu ra: Hãy tìm dãy con các phần tử liên tiếp của A có tổng lớn nhất.***

***Phương pháp: Theo phương pháp Quy hoạch động.***

***Lời giải:***

B1: Phân rã:

-Gọi MaxS[i] là tổng lớn nhất của dãy con liên tiếp có i phần tử a[1]…a[i].

-Khi đó MaxS[12] là giá trị lớn nhất của dãy con liên tiếp cần tìm.

B2: Bài toán cơ sở:

-Với i=1 ta có: MaxS[1] = a[1] = 3

B3: Tổng hợp:

Với: s: chỉ số đầu, e: chỉ số cuối, s1: chỉ số đầu tạm và:

Nếu MaxE[i-1] > 0 thì MaxE[i] = MaxE[i-1] + a[i]

Nếu MaxE[i-1] < 0 thì MaxE[i] = a[i]

MaxS[i] = max{MaxS[i-1], MaxE[i]}

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| i | a[i] | MaxE[i] | s1 | MaxS[i] | s | e | Dãy con tổng lớn nhất |
| 1 | 3 | 3 | 1 | 3 | 1 | 1 | 3 |
| 2 | 5 | 8 | 1 | 8 | 1 | 2 | 3, 5 |
| 3 | -9 | -1 | 1 | 8 | 1 | 2 | 3, 5 |
| 4 | 10 | 10 | 4 | 10 | 4 | 4 | 10 |
| 5 | 8 | 18 | 4 | 18 | 4 | 5 | 10, 8 |
| 6 | 6 | 24 | 4 | 24 | 4 | 6 | 10, 8, 6 |
| 7 | 4 | 28 | 4 | 28 | 4 | 7 | 10, 8, 6, 4 |
| 8 | -15 | 13 | 4 | 28 | 4 | 7 | 10, 8, 6, 4 |
| 9 | -6 | 7 | 4 | 28 | 4 | 7 | 10, 8, 6, 4 |
| 10 | -4 | 3 | 4 | 28 | 4 | 7 | 10, 8, 6, 4 |
| 11 | 1 | 4 | 4 | 28 | 4 | 7 | 10, 8, 6, 4 |
| 12 | 2 | 6 | 4 | 28 | 4 | 7 | 10, 8, 6, 4 |

* Dãy con cần tìm là: A[4]-A[7]: 10, 8, 6, 4

***VD3: Đầu vào: Mảng 10 số A[1….10] = 4, 9, 2, 7, 5, -4, 6, -5, -10, -9***

***Đầu ra: Hãy tìm dãy con các phần tử liên tiếp của A có tổng lớn nhất.***

***Phương pháp: Theo phương pháp Quy hoạch động.***

***Lời giải:***

B1: Phân rã:

-Gọi MaxS[i] là tổng lớn nhất của dãy con liên tiếp có i phần tử a[1]…a[i].

-Khi đó MaxS[10] là giá trị lớn nhất của dãy con liên tiếp cần tìm.

B2: Bài toán cơ sở:

-Với i=1 ta có: MaxS[1] = a[1] = 4

B3: Tổng hợp:

Với: s: chỉ số đầu, e: chỉ số cuối, s1: chỉ số đầu tạm và:

Nếu MaxE[i-1] > 0 thì MaxE[i] = MaxE[i-1] + a[i]

Nếu MaxE[i-1] < 0 thì MaxE[i] = a[i]

MaxS[i] = max{MaxS[i-1], MaxE[i]}

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| i | a[i] | MaxE[i] | s1 | MaxS[i] | s | e | Dãy con tổng lớn nhất |
| 1 | 4 | 4 | 1 | 4 | 1 | 1 | 4 |
| 2 | 9 | 13 | 1 | 13 | 1 | 2 | 4, 9 |
| 3 | 2 | 15 | 1 | 15 | 1 | 3 | 4, 9, 2 |
| 4 | 7 | 22 | 1 | 22 | 1 | 4 | 4, 9, 2, 7 |
| 5 | 5 | 27 | 1 | 27 | 1 | 5 | 4, 9, 2, 7, 5 |
| 6 | -4 | 23 | 1 | 27 | 1 | 5 | 4, 9, 2, 7, 5 |
| 7 | 6 | 29 | 1 | 29 | 1 | 7 | 4, 9, 2, 7, 5, -4, 6 |
| 8 | -5 | 24 | 1 | 29 | 1 | 7 | 4, 9, 2, 7, 5, -4, 6 |
| 9 | -10 | 14 | 1 | 29 | 1 | 7 | 4, 9, 2, 7, 5, -4, 6 |
| 10 | -9 | 5 | 1 | 29 | 1 | 7 | 4, 9, 2, 7, 5, -4, 6 |

* Dãy con cần tìm là: A[1]-A[7]: 4, 9, 2, 7, 5, -4, 6

***VD4: Đầu vào: Mảng 10 số A[1….10] = 6, 8, 3, 6, 1, -5, 3, -7, 0, 1***

***Đầu ra: Hãy tìm dãy con các phần tử liên tiếp của A có tổng lớn nhất.***

***Phương pháp: Theo phương pháp Quy hoạch động.***

***Lời giải:***

B1: Phân rã:

-Gọi MaxS[i] là tổng lớn nhất của dãy con liên tiếp có i phần tử a[1]…a[i].

-Khi đó MaxS[10] là giá trị lớn nhất của dãy con liên tiếp cần tìm.

B2: Bài toán cơ sở:

-Với i=1 ta có: MaxS[1] = a[1] = 6

B3: Tổng hợp:

Với: s: chỉ số đầu, e: chỉ số cuối, s1: chỉ số đầu tạm và:

Nếu MaxE[i-1] > 0 thì MaxE[i] = MaxE[i-1] + a[i]

Nếu MaxE[i-1] < 0 thì MaxE[i] = a[i]

MaxS[i] = max{MaxS[i-1], MaxE[i]}

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| i | a[i] | MaxE[i] | s1 | MaxS[i] | s | e | Dãy con tổng lớn nhất |
| 1 | 6 | 6 | 1 | 6 | 1 | 1 | 6 |
| 2 | 8 | 14 | 1 | 14 | 1 | 2 | 6, 8 |
| 3 | 3 | 17 | 1 | 17 | 1 | 3 | 6, 8, 3 |
| 4 | 6 | 23 | 1 | 23 | 1 | 4 | 6, 8, 3, 6 |
| 5 | 1 | 24 | 1 | 24 | 1 | 5 | 6, 8, 3, 6, 1 |
| 6 | -5 | 19 | 1 | 24 | 1 | 5 | 6, 8, 3, 6, 1 |
| 7 | 3 | 22 | 1 | 24 | 1 | 5 | 6, 8, 3, 6, 1 |
| 8 | -7 | 15 | 1 | 24 | 1 | 5 | 6, 8, 3, 6, 1 |
| 9 | 0 | 15 | 1 | 24 | 1 | 5 | 6, 8, 3, 6, 1 |
| 10 | 1 | 16 | 1 | 24 | 1 | 5 | 6, 8, 3, 6, 1 |

* Dãy con cần tìm là: A[1]-A[5]: 6, 8, 3, 6, 1

***VD5: Đầu vào: Mảng 10 số A[1….10] = 3, 7, 4, 8, -10, 6, -11, 5, 7, -2***

***Đầu ra: Hãy tìm dãy con các phần tử liên tiếp của A có tổng lớn nhất.***

***Phương pháp: Theo phương pháp Quy hoạch động.***

***Lời giải:***

B1: Phân rã:

-Gọi MaxS[i] là tổng lớn nhất của dãy con liên tiếp có i phần tử a[1]…a[i].

-Khi đó MaxS[10] là giá trị lớn nhất của dãy con liên tiếp cần tìm.

B2: Bài toán cơ sở:

-Với i=1 ta có: MaxS[1] = a[1] = 3

B3: Tổng hợp:

Với: s: chỉ số đầu, e: chỉ số cuối, s1: chỉ số đầu tạm và:

Nếu MaxE[i-1] > 0 thì MaxE[i] = MaxE[i-1] + a[i]

Nếu MaxE[i-1] < 0 thì MaxE[i] = a[i]

MaxS[i] = max{MaxS[i-1], MaxE[i]}

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| i | a[i] | MaxE[i] | s1 | MaxS[i] | s | e | Dãy con tổng lớn nhất |
| 1 | 3 | 3 | 1 | 3 | 1 | 1 | 3 |
| 2 | 7 | 10 | 1 | 10 | 1 | 2 | 3, 7 |
| 3 | 4 | 14 | 1 | 14 | 1 | 3 | 3, 7, 4 |
| 4 | 8 | 22 | 1 | 22 | 1 | 4 | 3, 7, 4, 8 |
| 5 | -10 | 12 | 1 | 22 | 1 | 4 | 3, 7, 4, 8 |
| 6 | 6 | 18 | 1 | 22 | 1 | 4 | 3, 7, 4, 8 |
| 7 | -11 | 9 | 1 | 22 | 1 | 4 | 3, 7, 4, 8 |
| 8 | 5 | 14 | 1 | 22 | 1 | 4 | 3, 7, 4, 8 |
| 9 | 7 | 21 | 1 | 22 | 1 | 4 | 3, 7, 4, 8 |
| 10 | -2 | 19 | 1 | 22 | 1 | 4 | 3, 7, 4, 8 |

* Dãy con cần tìm là: A[1]-A[4]: 3, 7, 4, 8

**IV.Chương trình sử dụng C, C++**

//Tim day con lien tiep co tong lon nhat theo quy hoach dong

#include<iostream>

#include<conio.h>

#include<stdio.h>

#include<stdlib.h>

using namespace std;

int n, a[100], MaxS, MaxE, s, e, s1;

void Nhap() {

cout << "\n Nhap n=";

cin >> n;

for (int i = 1;i <= n;i++)

{

cout << "\n Nhap gia tri phan tu thu "<<i<<":\n";

cout << "a[" << i << "] = ";

cin >> a[i];

}

}

void Xuat(){

cout<<"\nDay a: ";

for (int i=1; i<=n; i++)

{

cout <<a[i]<<",";

}

}

void Tim() {

MaxS=a[1];

MaxE=a[1];

s=1;

e=1;

s1=1;

for (int i=2; i<=n; i++)

{

if(MaxE>0)

{

MaxE=MaxE+a[i];

}

else

{

MaxE=a[i];

s1=i;

}

if(MaxE>MaxS)

{

MaxS=MaxE;

s=s1;

e=i;

}

}

cout<<"\nDay con can tim la: ";

for (int i=s; i<=e; i++)

{

cout <<a[i]<<",";

}

}

int main(){

Nhap();

Xuat();

Tim();

return 0;

}

**V.Đánh giá độ phức tạp thuật toán:**

Cài đặt chương trình như ở mục IV ta có:

Trong chương trình cài đặt có

**+5 phép gán:**

**MaxS=a[1];**

**MaxE=a[1];**

**s=1;**

**e=1;**

**s1=1;**

* Độ phức tạp của mỗi phép gán là hằng số: O(1)

***Theo Quy tắc cộng: T1(n)=O(f(n)), T2(n)=O(g(n))***

***=> T1(n)+T2(n)=O(f(n)+g(n))***

***Theo Quy tắc lấy max: T(n)=O(f(n)+g(n))=O(max(f(n), g(n)))***

Ta có độ phức tạp của phép gán trên là **T1(n)=O(1).**

**+4 vòng lặp for**

**VL1: for (int i = 1;i <= n;i++)**

**{**

**cout << "\n Nhap gia tri phan tu thu "<<i<<":\n";**

**cout << "a[" << i << "] = ";**

**cin >> a[i];**

**}**

Vòng lặp for thực hiện n lần với việc nhập nên độ phức tạp là O(n).

**VL2: for (int i=1; i<=n; i++)**

**{**

**cout <<a[i]<<",";**

**}**

Vòng lặp for thực hiện n lần với việc xuất nên độ phức tạp là O(n).

**VL3: for (int i=2; i<=n; i++)**

**{**

**if(MaxE>0)**

**{**

**MaxE=MaxE+a[i];**

**}**

**else**

**{**

**MaxE=a[i];**

**s1=i;**

**}**

**if(MaxE>MaxS)**

**{**

**MaxS=MaxE;**

**s=s1;**

**e=i;**

**}**

**}**

Trong vòng lặp for này có lồng **2 vòng lặp if**:

**Vòng lặp if 1: if(MaxE>0) => T0(n) = O(n)**

**{**

**MaxE=MaxE+a[i]; =>T1(n)=O(1)**

**}**

**else**

**{**

**MaxE=a[i]; =>T2(n)=O(1)**

**s1=i; =>T3(n)=O(1)**

**}**

Khi đó độ phức tạp thuật toán của vòng lặp if bằng:

T0(n)+max(T1(n),T2(n),T3(n)) = O(n) (theo quy tắc cộng và quy tắc lấy max)

**Vòng lặp if 2: if(MaxE>MaxS) => T0(n)=O(1)**

**{**

**MaxS=MaxE; => T1(n)=O(1)**

**s=s1; => T2(n)=O(1)**

**e=i; => T3(n)=O(1)**

**}**

Khi đó độ phức tạp thuật toán của vòng lặp if bằng:

T0(n)+max(T1(n),T2(n),T3(n)) = O(1) (theo quy tắc cộng và quy tắc lấy max)

Theo quy tắc cộng và quy tắc lấy max ta có độ phức tạp ở trong vòng lặp for là O(n).

***Với Quy tắc nhân: đoạn chương trình có thời gian thực hiện là T(n)=O(f(n)) nếu thực hiện k(n) lần đoạn chương trình với k(n)=O(g(n)) thì độ phức tạp là O(g(n).f(n))***

Áp dụng quy tắc nhân cho vòng lặp for với n vòng lặp => Độ phức tạp của vòng lặp for: O(n2)

**VL4: for (int i=s; i<=e; i++)**

**{**

**cout <<a[i]<<",";**

**}**

Vòng lặp for thực hiện e-s lần với việc xuất nên độ phức tạp là O(e-s) ((e-s)<n).

* Tổng hợp lại ta có 4 vòng lặp for với lần lượt độ phức tạp là: O(n), O(n), O(n2), O(e-s). Áp dụng quy tắc cộng và quy tắc lấy max ta có độ phức tạp của vòng lặp là: **T2(n)=O(n2)**
* ***Áp dụng quy tắc cộng và quy tắc lấy max cho T1(n) và T2(n) ta được độ phức tạp của cả chương trình là* O(n2).**