1. **GIỚI THIỆU CHUNG VỀ THUẬT TOÁN THAM LAM**
2. **Ý tưởng**

Phương pháp tham lam là kỹ thuật thiết kế thường được dung để giải các

bài toán tối ưu. Phương pháp được tiến hành trong nhiều bước. Tại mỗi bước, theo một lựa chọn nào đó ( xác định bằng một hàm chọn), sẽ tìm ra một lời giải tối ưu cho bài toán nhỏ tương ứng. Lời giải của bài toán được bổ sung dần từng bước từ lời giải của các bài toán con.

Lời giải được xây dựng như thế có chắc là lời giải tối ưu của bài toán?

Các lời giải tìm được bằng phương pháp tham lam thường là chấp nhận

được theo điều kiện nào đó, chưa chắc là tối ưu.

Cho trước một tập A gồm n đối tượng, ta cần phải chọn một tập con S của A. Với một tập con S được chọn ra thỏa mãn các yêu cầu của bài toán, ta gọi là một nghiệm chấp nhận được. Một hàm mục tiêu gắn mỗi nghiệm chấp nhận được với một giá trị. Nghiệm tối ưu là nghiệm chấp nhận được với giá trị nhỏ nhất ( lớn nhất).

Đặc trưng tham lam của phương pháp thể hiện bởi: trong mối bước việc xử lí sẽ tuân theo một sự lựa chọn trước, không kể đến tình trạng không tốt có thể xảy ra khi thực hiện lựa chọn lúc đầu.

1. **Mô hình**

Chọn S từ tập A.

Tính chất tham lam của thuật toán định hướng bởi hàm Chọn.

* Khởi động S = ∅;
* Trong khi A ≠ ∅:
  + Chọn phần tử tốt nhất của A gán vào x: x= Chọn(A);
  + Cập nhật các đối tượng để chọn: A = A – {x};
  + Nếu S∪{x} thỏa mãn yêu cầu bài toán thì
    - Cập nhật lời giải: S = S∪{x};

Thủ tục thuật toán tham lam có thể cài đặt như sau:

Input A[1..n]

Output S//lời giải

Greedy (A,n) ≡ S = ∅;

while ( A ≠ ∅)

{

x = Chọn(A);

A = A-{x};

If(S∪{x}chấp nhận được)

S = S∪{x};

}

Return S;

1. **THUẬT TOÁN THAM LAM GIẢI BÀI TOÁN NGƯỜI DU LỊCH**
2. **Bài toán**

Một người du lịch muốn tham quan n thành phố T1, …, Tn. Xuất phát từ một thành phố nào đó, người du lịch muốn đi qua tất cả các thành phố còn lại, mỗi thành phố đi qua đúng 1 lần rồi quay trở lại thành phố xuất phát.

Gọi Cij là chi phí đi từ thành phố Ti đến Tj. Hãy tìm một hành trình thỏa yêu cầu bài toán sao cho chi phí là nhỏ nhất.

1. **Mô tả thuật toán:**
2. Ý tưởng:

Đây là bài toán tìm chu trình có trọng số nhỏ nhất trong một đơn đồ thị có hướng có trọng số. Thuật toán tham lam cho bài toán là chọn thành phố có chi phí nhỏ nhất tính từ thành phố hiện thời đến các thành phố chưa qua

1. Mô tả chi tiết thuật toán:

Input: G = (V,E), u (điểm bắt đầu)

Output: TOUR // Hành trình tối ưu

Mô tả :

COST;//Chi phí tương ứng

TOUR = 0; COST = 0; v = u; // Khởi tạo

Mọi k = 1 -> n ://Thăm tất cả các thành phố

// Chọn cạnh kề

- Chọn <v, w> là đoạn nối 2 thành phố có chi phí nhỏ nhất tính từ thành phố v đến các thành phố chưa qua.

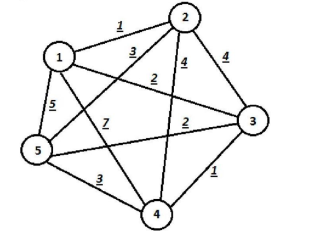
- TOUR = TOUR + <v, w>; //Cập nhật lời giải

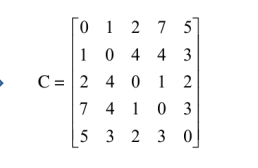
- COST = COST + Cvw ; //Cập nhật chi phí

// Chuyến đi hoàn thành TOUR := TOUR + <v, w>; COST := COST + Cvw

# Thực hiện các bước của thuật toán

Bộ dữ liệu 1: n =5 ,u=1





Hoạt động của thuật toán:

|  |  |  |
| --- | --- | --- |
| Bước | TOUR | COST |
| 1 | 1 | 0 |
| 2 | 1-> 2 | 1 |
| 3 | 1-> 2-> 5 | 4 |
| 4 | 1-> 2-> 5->3 | 6 |
| 5 | 1-> 2-> 5->3->4 | 7 |

Bộ dữ liệu 2: n = 10, u = 2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 2 | 1 | 7 | 1 | 2 | 6 | 5 | 7 | 4 |
| 2 | 0 | 4 | 2 | 3 | 5 | 8 | 1 | 8 | 9 |
| 1 | 4 | 0 | 6 | 9 | 1 | 8 | 2 | 6 | 8 |
| 7 | 2 | 6 | 0 | 5 | 2 | 3 | 9 | 4 | 3 |
| 1 | 3 | 9 | 5 | 0 | 6 | 4 | 7 | 2 | 2 |
| 2 | 5 | 1 | 2 | 6 | 0 | 4 | 3 | 1 | 6 |
| 6 | 8 | 8 | 3 | 4 | 4 | 0 | 2 | 7 | 5 |
| 5 | 1 | 2 | 9 | 7 | 3 | 2 | 0 | 2 | 3 |
| 7 | 8 | 6 | 4 | 2 | 1 | 7 | 2 | 0 | 5 |
| 4 | 9 | 8 | 3 | 2 | 6 | 5 | 3 | 5 | 0 |

Hoạt động của thuật toán:

|  |  |  |
| --- | --- | --- |
| Bước | TOUR | COST |
| 1 | 2 | 0 |
| 2 | 2 -> 8 | 1 |
| 3 | 2 -> 8 -> 3 | 3 |
| 4 | 2 -> 8 -> 3 -> 1 | 4 |
| 5 | 2 -> 8 -> 3 -> 1 -> 5 | 5 |
| 6 | 2 -> 8 -> 3 -> 1 -> 5 -> 9 | 7 |
| 7 | 2 -> 8 -> 3 -> 1 -> 5 -> 9 -> 6 | 8 |
| 8 | 2 -> 8 -> 3 -> 1 -> 5 -> 9 -> 6 -> 4 | 10 |
| 9 | 2 -> 8 -> 3 -> 1 -> 5 -> 9 -> 6 -> 4 -> 7 | 13 |
| 10 | 2 -> 8 -> 3 -> 1 -> 5 -> 9 -> 6 -> 4 -> 7 -> 10 | 18 |
| 11 | 2 -> 8 -> 3 -> 1 -> 5 -> 9 -> 6 -> 4 -> 7 -> 10 -> 2 | 27 |

Bộ dữ liệu 3: n = 10, u = 5

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 3 | 5 | 2 | 6 | 1 | 6 | 3 | 2 | 7 |
| 3 | 0 | 6 | 7 | 3 | 1 | 5 | 7 | 1 | 7 |
| 5 | 6 | 0 | 5 | 4 | 6 | 5 | 6 | 3 | 2 |
| 2 | 7 | 5 | 0 | 1 | 3 | 7 | 6 | 2 | 6 |
| 6 | 3 | 4 | 1 | 0 | 2 | 1 | 3 | 2 | 1 |
| 1 | 1 | 6 | 3 | 2 | 0 | 7 | 4 | 5 | 6 |
| 6 | 5 | 5 | 7 | 1 | 7 | 0 | 8 | 4 | 3 |
| 3 | 7 | 6 | 6 | 3 | 4 | 8 | 0 | 2 | 5 |
| 2 | 1 | 3 | 2 | 2 | 5 | 4 | 2 | 0 | 7 |
| 7 | 7 | 2 | 6 | 1 | 6 | 3 | 5 | 7 | 0 |

Hoạt động của thuật toán:

|  |  |  |
| --- | --- | --- |
| Bước | TOUR | COST |
| 1 | 5 | 0 |
| 2 | 5 -> 4 | 1 |
| 3 | 5 -> 4 -> 1 | 3 |
| 4 | 5 -> 4 -> 1 -> 6 | 4 |
| 5 | 5 -> 4 -> 1 -> 6 -> 2 | 5 |
| 6 | 5 -> 4 -> 1 -> 6 -> 2 -> 9 | 6 |
| 7 | 5 -> 4 -> 1 -> 6 -> 2 -> 9 -> 8 | 8 |
| 8 | 5 -> 4 -> 1 -> 6 -> 2 -> 9 -> 8 -> 10 | 13 |
| 9 | 5 -> 4 -> 1 -> 6 -> 2 -> 9 -> 8 -> 10 -> 3 | 15 |
| 10 | 5 -> 4 -> 1 -> 6 -> 2 -> 9 -> 8 -> 10 -> 3 -> 7 | 20 |
| 11 | 5 -> 4 -> 1 -> 6 -> 2 -> 9 -> 8 -> 10 -> 3 -> 7 -> 5 | 21 |

Bộ dữ liệu 4: n = 12, u = 11

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 8 | 3 | 7 | 2 | 6 | 5 | 3 | 4 | 2 | 3 |
| 1 | 0 | 4 | 2 | 2 | 8 | 7 | 8 | 5 | 9 | 1 | 4 |
| 8 | 4 | 0 | 6 | 9 | 7 | 8 | 2 | 1 | 7 | 4 | 7 |
| 3 | 2 | 6 | 0 | 5 | 2 | 1 | 1 | 4 | 3 | 5 | 8 |
| 7 | 2 | 9 | 5 | 0 | 6 | 4 | 7 | 6 | 1 | 6 | 9 |
| 2 | 8 | 7 | 2 | 6 | 0 | 4 | 3 | 1 | 2 | 7 | 2 |
| 6 | 7 | 8 | 1 | 4 | 4 | 0 | 2 | 1 | 5 | 3 | 7 |
| 5 | 8 | 2 | 1 | 7 | 3 | 2 | 0 | 2 | 3 | 5 | 1 |
| 3 | 5 | 1 | 4 | 6 | 1 | 1 | 2 | 0 | 5 | 2 | 8 |
| 4 | 9 | 7 | 3 | 1 | 2 | 5 | 3 | 2 | 0 | 8 | 3 |
| 2 | 1 | 4 | 5 | 6 | 7 | 3 | 5 | 2 | 8 | 0 | 6 |
| 3 | 4 | 7 | 8 | 9 | 2 | 7 | 1 | 8 | 3 | 6 | 0 |

Hoạt động của thuật toán:

|  |  |  |
| --- | --- | --- |
| Bước | TOUR | COST |
| 1 | 11 | 0 |
| 2 | 11 -> 2 | 1 |
| 3 | 11 -> 2 -> 1 | 2 |
| 4 | 11 -> 2 -> 1 -> 6 | 4 |
| 5 | 11 -> 2 -> 1 -> 6 -> 9 | 5 |
| 6 | 11 -> 2 -> 1 -> 6 -> 9 -> 3 | 6 |
| 7 | 11 -> 2 -> 1 -> 6 -> 9 -> 3 -> 8 | 8 |
| 8 | 11 -> 2 -> 1 -> 6 -> 9 -> 3 -> 8 -> 4 | 9 |
| 9 | 11 -> 2 -> 1 -> 6 -> 9 -> 3 -> 8 -> 4 -> 7 | 10 |
| 10 | 11 -> 2 -> 1 -> 6 -> 9 -> 3 -> 8 -> 4 -> 7 -> 5 | 14 |
| 11 | 11 -> 2 -> 1 -> 6 -> 9 -> 3 -> 8 -> 4 -> 7 -> 5 -> 10 | 15 |
| 12 | 11 -> 2 -> 1 -> 6 -> 9 -> 3 -> 8 -> 4 -> 7 -> 5 -> 10->12 | 18 |
| 13 | 11 -> 2 -> 1 -> 6 -> 9 -> 3 -> 8 -> 4 -> 7 -> 5 -> 10->12-> 11 | 24 |

Bộ dữ liệu 5: n = 11, u = 2

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 3 | 1 | 7 | 8 | 2 | 6 | 5 | 3 | 4 | 3 |
| 3 | 0 | 4 | 2 | 2 | 8 | 7 | 8 | 5 | 9 | 4 |
| 1 | 4 | 0 | 6 | 9 | 7 | 8 | 2 | 1 | 7 | 7 |
| 7 | 2 | 6 | 0 | 5 | 2 | 1 | 1 | 4 | 3 | 8 |
| 8 | 2 | 9 | 5 | 0 | 6 | 4 | 7 | 6 | 1 | 9 |
| 2 | 8 | 7 | 2 | 6 | 0 | 4 | 3 | 1 | 2 | 2 |
| 6 | 7 | 8 | 1 | 4 | 4 | 0 | 2 | 1 | 5 | 7 |
| 5 | 8 | 2 | 1 | 7 | 3 | 2 | 0 | 2 | 3 | 1 |
| 3 | 5 | 1 | 4 | 6 | 1 | 1 | 2 | 0 | 5 | 8 |
| 4 | 9 | 7 | 3 | 1 | 2 | 5 | 3 | 2 | 0 | 3 |
| 3 | 4 | 7 | 8 | 9 | 2 | 7 | 1 | 8 | 3 | 0 |

Hoạt động của thuật toán:

|  |  |  |
| --- | --- | --- |
| Bước | TOUR | COST |
| 1 | 2 | 0 |
| 2 | 2 -> 4 | 2 |
| 3 | 2 -> 4 -> 7 | 3 |
| 4 | 2 -> 4 -> 7 ->9 | 4 |
| 5 | 2 -> 4 -> 7 ->9->3 | 5 |
| 6 | 2 -> 4 -> 7 ->9->3 ->1 | 6 |
| 7 | 2 -> 4 -> 7 ->9->3>1 -> 6 | 8 |
| 8 | 2 -> 4 -> 7 -> 9 -> 3 -> 1 -> 6 -> 10 | 10 |
| 9 | 2 -> 4 -> 7 -> 9 -> 3 -> 1 -> 6 -> 10 -> 5 | 11 |
| 10 | 2 -> 4 -> 7 -> 9 -> 3 -> 1 -> 6 -> 10 -> 5 -> 8 | 18 |
| 11 | 2 -> 4 -> 7 -> 9 -> 3 -> 1 -> 6 -> 10 -> 5 -> 8 -> 11 | 19 |
| 12 | 2 -> 4 -> 7 -> 9 -> 3 -> 1 -> 6 -> 10 -> 5 -> 8 -> 11 ->2 | 23 |

# Cài đặt thuật toán

int GTS (mat a, int n, int TOUR[max], int Ddau)

{

//Khoi tao

int v, //Dinh dang xet

k, //Duyet qua n dinh de chon

w; //Dinh duoc chon trong moi buoc

int mini; //Chon min cac canh(cung) trong moi buoc

int COST; //Trong so nho nhat cua chu trinh

int daxet[max]; //Danh dau cac dinh da duoc su dung

for(k = 1; k <= n; k++)

daxet[k] = 0; //Chua dinh nao duoc xet

COST = 0; //Luc dau, gia tri COST == 0

int i; // Bien dem, dem tim du n dinh thi dung

v = Ddau; //Chon dinh xuat phat la 1

i = 1;

TOUR[i] = v; //Dua v vao chu trinh

daxet[v] = 1; //Dinh v da duoc xet

//Tim dinh gan nhat voi dinh dang xet

while(i < n)

{

mini = VC;

for (k = 1; k <= n; k++)

{

if(!daxet[k])

{

if(mini > a[v][k])

{

mini = a[v][k];

w = k;

}

}

}

v = w;

i++;

TOUR[i] = v;

daxet[v] = 1;

COST += mini;

}

//Tro ve dinh dau

COST += a[v][Ddau];

return COST;

}

# Độ phức tạp thuật toán

Thao tác chọn đỉnh thích hợp trong n đỉnh được tổ chức bằng một vòng lặp

để duyệt. Nên chi phí cho thuật toán xác định bởi 2 vòng lặp lồng nhau, nên

T(n) € O (n2).