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 cácbài 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 KRUSKAL – TÌM CÂY BAO TRÙM NHỎ NHẤT**
2. **Bài toán**

G = (V,E) là đơn đồ thị vô hướng liên thông, có trọng số.

V = {1,..,n} là tập các đỉnh. E là tập các cạnh (edge).

Một cây T gọi là cây bao trùm của G nếu T là đồ thị con của G và chứa mọi đỉnh của G.

Vấn đề là tìm cây bao trùm có trọng số nhỏ nhất: MST (Minimal Spanning Tree) của G.

Các thuật toán cơ bản giải bài toán trên là các thuật toán Prim và Ksuskal. Trong phần này, ta giới thiệu thuật toán Kruskal.

1. **Ý tưởng**

Thuật toán Kruskal xây dựng một đồ thị con T của G như sau:

* T = G(Vt,Et) là cây khung tối thiểu cần tìm
* Khi G có n đỉnh thì T có n-1 cạnh
* Ý tưởng (tham lam): Xây dựng tập n-1 cạnh của T theo nguyên tắc:

+ Khởi tạo Ej={}, VT = V

+ Xét lần lượt các cạnh có trọng số nhỏ đến lớn nếu không tạo thành chu trình trong T thì thêm cạnh đó vào VT.

1. **Mô tả thuật toán**

- Mô tả G bằng ma trận trọng số cạnh A[ij].

- D là mảng 1 chiều, nếu D[i]=k thì đỉnh i thuộc vào cây thứ k, D[i] = 0 thì đỉnh i chưa thuộc vào cây nào hết.

- Tìm min {A[i][j]} j = l..n, i =l..n trừ các cạnh (i,j) mà D[i]=D[j]<>0 (những cạnh đó tạo thành chu trình nên không thoả mãn). (bước 2)

- Thêm cạnh vừa tìm vào cây X lặp lại bước 2 khi T còn là rừng.

**\* Xử lý cạnh (i,j) khi được thêm vào T:**

- Nếu D[i]=D[j]=0, cạnh (i,j) chưa thuộc vào cây nên khi lấy 2 đỉnh này vào tập cạnh ta cho chúng thuộc vào 1 cây mới. Khi đó k=k+l và D[i]=D[j]=k.

- Nếu D[i]=0 và D[j]<>0: i chưa thuộc vào X j thuộc T => Ghép i vào cùng cây chứa j, D[i]=D[j].

- Nếu D[i]<>0 và D[j]=0: i thuộc vào X j không thuộc T => Ghép j vào cùng cây chứa i, D[j]=D[i].

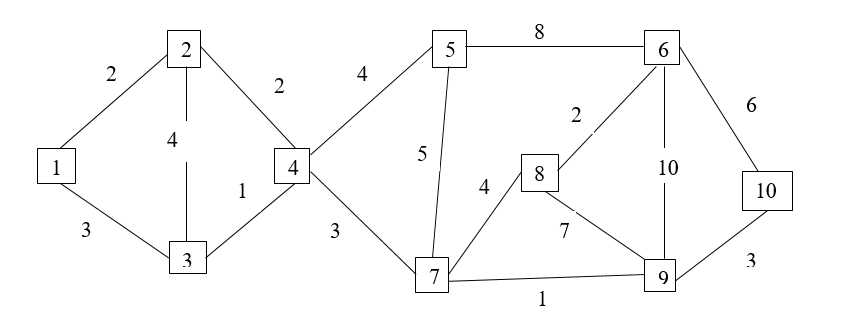
- Nếu D[i]oD[j] và D[i]<>0, D[j]<>0: i, j thuộc 2 cây khác nhau trong T => Ghép 2 cây thành 1.

1. **Thực hiện các bước của thuật toán**

***Bộ dữ liệu 1:***

8

1



**\* Các bước của thuật toán:**

- Bước 1: (Khởi tạo)

+ D[i]=0; với i từ 1 đến 10: Các đỉnh chưa thuộc cây nào

+ T=0: Chưa có cây nào

+ Dem=0: Cây cần tìm chưa có cạnh nào, cây cần tìm có n-1=9 cạnh

- Bước 2: (bước lặp)

**+ B1:** Tìm cạnh (i,j) có trọng số nhỏ nhất 🡪 cạnh (4-3): Trọng số là 1

Vì D[4]=D[3]=0 nên cho đỉnh 3,4 vào cây 1; **T=1;**

Cho D[4]=D[3]=1;

Sau bước này tăng biến Dem lên 1, vẫn nhỏ hơn n-1=9;

**+ B2:** Tìm cạnh có trọng số nhỏ nhất tiếp theo: Cạnh (7-9): Trọng số 1

Vì D[7]=D[9]=0 nên cho đỉnh 7,9 vào cây 2; **T=2;**

Cho D[7]=D[9]=2;

Sau bước này tăng biến Dem lên 2, vẫn nhỏ hơn n-1=9;

**+ B3:** Tìm cạnh có trọng số nhỏ nhất tiếp theo: Cạnh (1-2): Trọng số 2

Vì D[1]=D[2]=0 nên cho đỉnh 1,2 vào cây 3; **T=3;**

Cho D[1]=D[2]=3;

Sau bước này tăng biến Dem lên 3, vẫn nhỏ hơn n-1=9;

**+ B4:** Tìm cạnh có trọng số nhỏ nhất tiếp theo: Cạnh (2-4): Trọng số 2

Vì D[2]=3; D[4]=1 đều khác 0 và khác nhau (2 đỉnh 2,4 thuộc 2 cây khác nhau); Lúc này ta gộp 2 cây vào một ;

**T=2** (vì gộp cây nên T giảm 1); Cụ thể gộp vào cùng cây 1, tức cho D[2]=D[4]=1; Và cho luôn D[1]=1 vì đỉnh 1 cùng cây với đỉnh 2

Sau bước này tăng biến Dem lên 4, vẫn nhỏ hơn n-1=9;

**+ B5**: Tìm cạnh có trọng số nhỏ nhất tiếp theo: Cạnh (6-8): Trọng số 2

Vì D[6]=D[8]=0 nên cho đỉnh 6,8 vào cây 3; **T=3**;

Cho D[6]=D[8]=3;

Sau bước này tăng biến Dem lên 5, vẫn nhỏ hơn n-1=9;

**+ B6**: Tìm cạnh có trọng số nhỏ nhất tiếp theo: Cạnh (4-7): Trọng số 3

Vì D[4]=1; D[7]=2 đều khác 0 và khác nhau (2 đỉnh 4,7 thuộc 2 cây khác nhau); Lúc này ta gộp 2 cây vào một ; **T=2** (vì gộp cây nên T giảm 1); Cụ thể gộp vào cùng cây 1, tức cho D[7]=D[4]=1; Và cho luôn D[9]=1 vì đỉnh 9 cùng cây với đỉnh 7;

Sau bước này tăng biến Dem lên 6, vẫn nhỏ hơn n-1=9;

**+ B7**: Tìm cạnh có trọng số nhỏ nhất tiếp theo: Cạnh (9-10): Trọng số 3

Vì D[9]=1; D[10]=0 nên cho đỉnh 10 vào cây chứa đỉnh 9 (cây 1); **T=2** (vì không có cây mới nên T giữ nguyên); D[10]=1;

Sau bước này tăng biến Dem lên 7, vẫn nhỏ hơn n-1=9;

**+ B8**: Tìm cạnh có trọng số nhỏ nhất tiếp theo: Cạnh (4-5): Trọng số 4

Vì D[4]=1; D[5]=0 nên cho đỉnh 5 vào cây chứa đỉnh 4 (cây 1); **T=2** (vì không có cây mới nên T giữ nguyên); D[5]=1;

Sau bước này tăng biến Dem lên 8, vẫn nhỏ hơn n-1=9;

**+ B9**: Tìm cạnh có trọng số nhỏ nhất tiếp theo: Cạnh (7-8): Trọng số 4

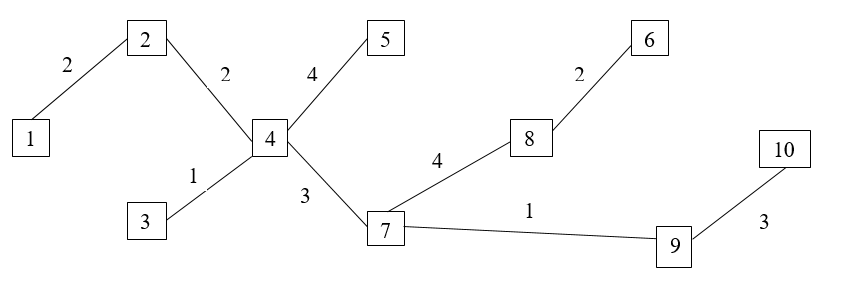
Vì D[7]=1; D[8]=3 đều khác 0 và khác nhau (2 đỉnh 7,8 thuộc 2 cây khác nhau); Lúc này ta gộp 2 cây vào một; **T=1** (vì gộp cây nên T giảm 1); Cụ thể gộp vào cùng cây 1, tức cho D[8]=D[7]=1; Và cho luôn D[6]=1 vì đỉnh 6 cùng cây với đỉnh 8;

Sau bước này tăng biến Dem lên 8, vẫn nhỏ hơn n-1=9;

**+ B10**: Dem=9=n-1: Số cạnh của cây bao trùm đủ, thuật toán kết thúc

**🡪 T=1; D[i]=1 với i chạy từ 1 đến 10; Đây cũng chính là cây bao trùm nhỏ nhất cần tìm**

**\* Đồ thị kết quả:**

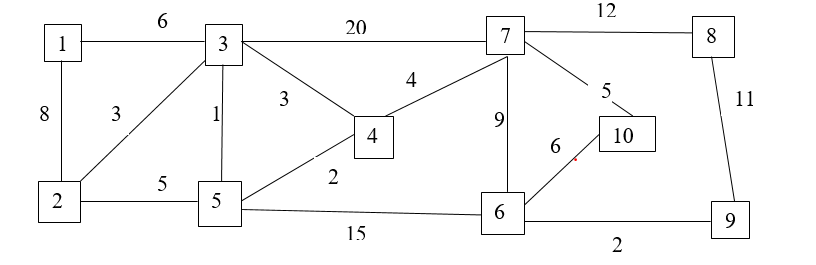


**\* Bảng tóm tắt:**

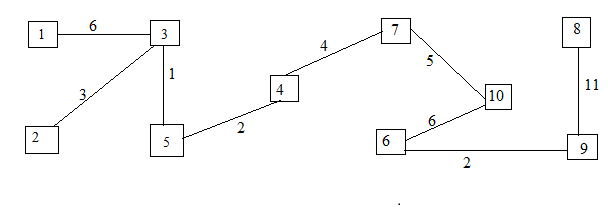
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Bước | Các cạnh chọn | Trọng số | Dem(Dừng khi Dem=n-1) | T | Đỉnh đã duyệt |
| 0 | - |  | 0 | 0 |  |
| 1 | (4,3) | 1 | 1 | 1 | 3,4 |
| 2 | (7,9) | 1 | 2 | 2 | 3,4,7,9 |
| 3 | (1,2) | 2 | 3 | 3 | 1,2,3,4,7,9 |
| 4 | (2,4) | 2 | 4 | 2 | 1,2,3,4,7,9 |
| 5 | (6,8) | 2 | 5 | 3 | 1,2,3,4,6,7,8,9 |
| 6 | (4,7) | 3 | 6 | 2 | 1,2,3,4,6,7,8,9 |
| 7 | (9,10) | 3 | 7 | 2 | 1,2,3,4,6,7,8,9,10 |
| 8 | (4,5) | 4 | 8 | 2 | 1,2,3,4,5,6,7,8,9,10 |
| 9 | (7,8) | 4 | 9 | 1 | 1,2,3,4,5,6,7,8,9,10 |

🡪 Trọng số của cây khung nhỏ nhất là: 1+1+2+2+2+3+3+4+4=22

***Bộ dữ liệu 2:***



**\* Thực hiện thuật toán Kruskal, ta được MST:**

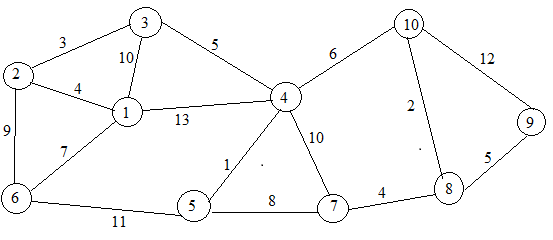


**\* Hoạt động của thuật toán:**

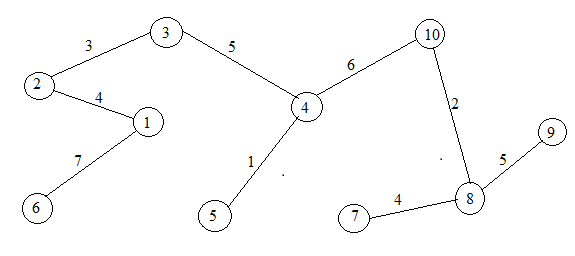
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Bước | Các cạnh chọn | Trọng số | Dem(Dừng khi Dem=n-1) | T | Đỉnh đã duyệt |
| 0 | - |  | 0 | 0 |  |
| 1 | (3,5) | 1 | 1 | 1 | 3,5 |
| 2 | (4,5) | 2 | 2 | 1 | 3,4,5 |
| 3 | (6-9) | 2 | 3 | 2 | 3,4,5,6,9 |
| 4 | (2,3) | 3 | 4 | 2 | 2,3,4,5,6,9 |
| 5 | (4,7) | 4 | 5 | 2 | 2,3,4,5,6,7,9 |
| 6 | (7,10) | 5 | 6 | 2 | 2,3,4,5,6,7,9,10 |
| 7 | (1,3) | 6 | 7 | 2 | 1,2,3,4,5,6,7,9,10 |
| 8 | (6,10) | 6 | 8 | 1 | 1,2,3,4,5,6,7,9,10 |
| 9 | (8,9) | 11 | 9 | 1 | 1,2,3,4,5,6,7,8,9,10 |

🡪 Trọng số của cây khung nhỏ nhất là: 1+2+2+2+3+4+5+5+6=30

***Bộ dữ liệu 3:***

******

**\* Thực hiện thuật toán Prim bắt đầu từ đỉnh 1 ta được MST:**

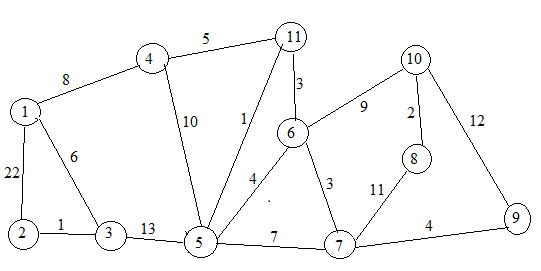


**\* Hoạt động của thuật toán:**

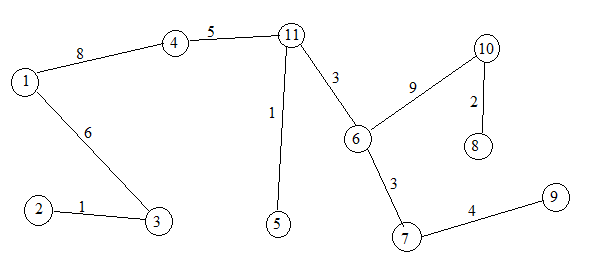
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Bước | Các cạnh chọn | Trọng số | Dem(Dừng khi Dem=n-1) | T | Đỉnh đã duyệt |
| 0 | - |  | 0 | 0 |  |
| 1 | (4,5) | 1 | 1 | 1 | 4,5 |
| 2 | (8,10) | 2 | 2 | 2 | 4,5,8,10 |
| 3 | (2,3) | 3 | 3 | 3 | 2,3,4,5,8,10 |
| 4 | (1,2) | 4 | 4 | 3 | 1,2,3,4,5,8,10 |
| 5 | (7,8) | 4 | 5 | 3 | 1,2,3,4,5,7,8,10 |
| 6 | (3,4) | 5 | 6 | 2 | 1,2,3,4,5,7,8,10 |
| 7 | (8,9) | 5 | 7 | 2 | 1,2,3,4,5,7,8,9,10 |
| 8 | (4,10) | 6 | 8 | 1 | 1,2,3,4,5,7,8,9,10 |
| 9 | (1,6) | 7 | 9 | 1 | 1,2,3,4,5,6,7,8,9,10 |

🡪 Trọng số của cây khung nhỏ nhất là: 1+2+3+4+4+5+5+6+7=37

***Bộ dữ liệu 4:***

******

**\* Thực hiện thuật toán Prim bắt đầu từ đỉnh 1 ta được MST:**

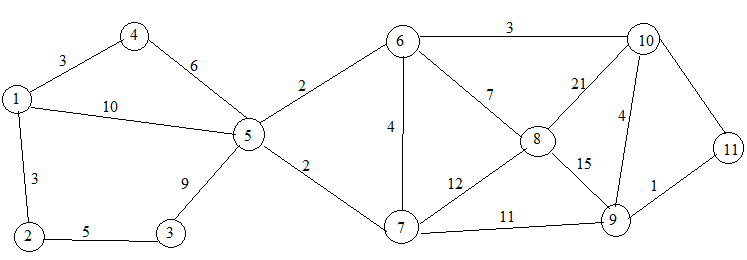
******

**\* Hoạt động của thuật toán**

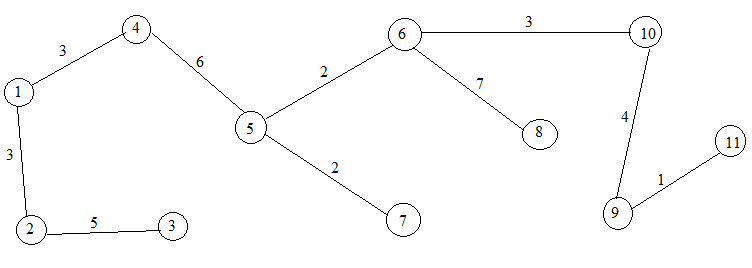
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Các cạnh chọn | Trọng số | Dem(Dừng khi Dem=n-1) | T | Đỉnh đã duyệt |
| - |  | 0 | 0 |  |
| (2,3) | 1 | 1 | 1 | 2,3 |
| (5,11) | 1 | 2 | 2 | 2,3,5,11 |
| (8,10) | 2 | 3 | 3 | 2,3,5,8,11 |
| (6,7) | 3 | 4 | 4 | 2,3,5,6,7,8,10,11 |
| (6,11) | 3 | 5 | 3 | 2,3,5,6,7,8,10,11 |
| (7,9) | 4 | 6 | 3 | 2,3,5,6,7,8,9,10,11 |
| (4,11) | 5 | 7 | 3 | 2,3,4,5,6,7,8,9,10,11 |
| (1,3) | 6 | 8 | 3 | 1,2,3,4,5,6,7,8,9,10,11 |
| (1,4) | 8 | 9 | 2 | 1,2,3,4,5,6,7,8,9,10,11 |
| (6,10) | 9 | 10 | 1 | 1,2,3,4,5,6,7,8,9,10,11 |

🡪 Trọng số của cây khung nhỏ nhất là: 1+1+2+3+3+4+5+6+8+9=42

***Bộ dữ liệu 5:***

******

**\* Thực hiện thuật toán Prim bắt đầu từ đỉnh 1 ta được MST:**



***\** Hoạt động của thuật toán**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Bước | Các cạnh chọn | Trọng số | Dem(Dừng khi Dem=n-1) | T | Đỉnh đã duyệt |
| 0 | - |  | 0 | 0 |  |
| 1 | (9,11) | 1 | 1 | 1 | 9,11 |
| 2 | (5,6) | 2 | 2 | 2 | 5,6,9,11 |
| 3 | (5,7) | 2 | 3 | 2 | 5,6,7,9,11 |
| 4 | (1,2) | 3 | 4 | 3 | 1,2,5,6,7,9,11 |
| 5 | (1,4) | 3 | 5 | 3 | 1,2,4,5,6,7,9,11 |
| 6 | (6,10) | 3 | 6 | 3 | 1,2,4,5,6,7,9,10,11 |
| 7 | (9,10) | 4 | 7 | 2 | 1,2,4,5,6,7,9,10,11 |
| 8 | (2,3) | 5 | 8 | 2 | 1,2,3,4,5,6,7,9,10,11 |
| 9 | (4,5) | 6 | 9 | 1 | 1,2,3,4,5,6,7,9,10,11 |
| 10 | (6,8) | 7 | 10 | 1 | 1,2,3,4,5,6,7,8,9,10,11 |

🡪 Trọng số của cây khung nhỏ nhất là: 1+2+2+3+3+3+4+5+6+7=36

1. **Độ phức tạp của thuật toán**