THIẾT KẾ THUẬT TOÁN 3

Một số kỹ thuật thiết kế

1. Kỹ thuật thiết kế tuần tự
2. Kỹ thuật thiết kế theo cấu trúc
3. Kỹ thuật đệ quy
4. Kỹ thuật tham lam
5. Kỹ thuật chia để trị
6. Kỹ thuật liệt kê
7. Lựa chọn cấu trúc dữ liệu thích hợp
8. Bài tập

4. Kỹ thuật tham lam

Ý tưởng:

- **Gốc của ý tưởng tham lam là "loang". Loang dần trong phạm vi có thể và cho đến khi gặp được lời giải hoặc hết phạm vi để "loang".**
- **Kỹ thuật "loang" là <u>tìm kiếm phần tử kế tiếp trong phạm vi "loang“</u>:**

  Tìm phần tử bên cạnh (lân cận, láng giềng gần nhất, kề), hoặc tùy theo mục tiêu bài toán tối ưu mà tìm phần tử nhỏ nhất, lớn nhất, bậc cao nhất, hoặc có thể thêm ràng buộc khác như không sinh ra chu trình, không lặp lại.

4. Kỹ thuật tham lam

Một số kiến thức bổ trợ về đồ thị:

- **Đồ thị G = {V,E}, V là tập hữu hạn khác rỗng, gọi là tập đỉnh và E là tập con của tập tích đề-các V×V. Nếu mỗi e=(u,v) ∈E có kể đến thứ tự thì G gọi là có hướng, ngược lại, G gọi là vô hướng.**
- **Với mọi *e*=(*u*,*v*)∈E, *e* gọi là liên thuộc với *u*, *v*; hai đỉnh *u*, *v* gọi là liền kề.**
- **Dãy các đỉnh {u_1, u_2, ..., u_m} được gọi là đường đi nối u_1 và u_m, và kí hiệu $\alpha(u_1,u_m)$, nếu mọi cặp đỉnh đứng cạnh nhau trong dãy là liền kề.**
- **G liên thông nếu giữa hai đỉnh bất kỳ của đồ thị tồn tại ít nhất một đường đi. Với đồ thị có hướng, liên thông, nếu giữa hai đỉnh (u,v) bất kỳ có cả hai đường đi $\alpha(u,v)$ và $\alpha(v,u)$ thì G được gọi là liên thông mạnh; ngược lại, gọi là liên thông yếu.**

4. Kỹ thuật tham lam

<u>Một số kiến thức bổ trợ về đồ thị</u>:

Cho đồ thị G ={V,E} và E' $\subseteq$ E.

- **Đồ thị T = {V, E'}được gọi là cây khung của G nếu T liên thông và không có chu trình.**
- **Tính chất:**
  - **G *có cây khung* T *khi và chỉ khi* G *liên thông*.**
  - **Đồ thị có *n* đỉnh, cây khung sẽ có *n*-1 cạnh.**

4. Kỹ thuật tham lam

Một số kiến thức bổ trợ về đồ thị:

Đồ thị G = {V,E} liên thông, V = {1, 2, ..., *n*}, được cho bởi danh sách cạnh

```
struct edge
  {       tên đỉnh        d,c;
          trọng số        t;
  }
edge  G[m];
```

4. Kỹ thuật tham lam

Một số kiến thức bổ trợ về đồ thị:

Đồ thị G = {V,E} liên thông, V = {1, 2, ..., *n*}, được cho bởi danh sách cạnh

```
struct edge
    {       tên đỉnh        d,c;
            trọng số        t;
    }
edge  G[m];
```

4. Kỹ thuật tham lam

Ví dụ 4.1:

Đồ thị G = {V,E} liên thông, V = {1, 2, ..., *n*}, được cho bởi danh sách cạnh

```
struct edge
   {        tên đỉnh        d,c;
            trọng số        t;
   }
edge  G[m];
```

Tìm cây khung tối tiểu (MST- minimum spanning tree) của G.

4. Kỹ thuật tham lam

Thuật toán Kruskal:

input $V=\{1, 2, ..., n\}$, G[1..m] of egde

output $V=\{1, 2, ..., n\}$, K[1..n-1] of egde

Ý tưởng:

Kí hiệu F là rừng sinh ra khi thực hiện thuật toán,

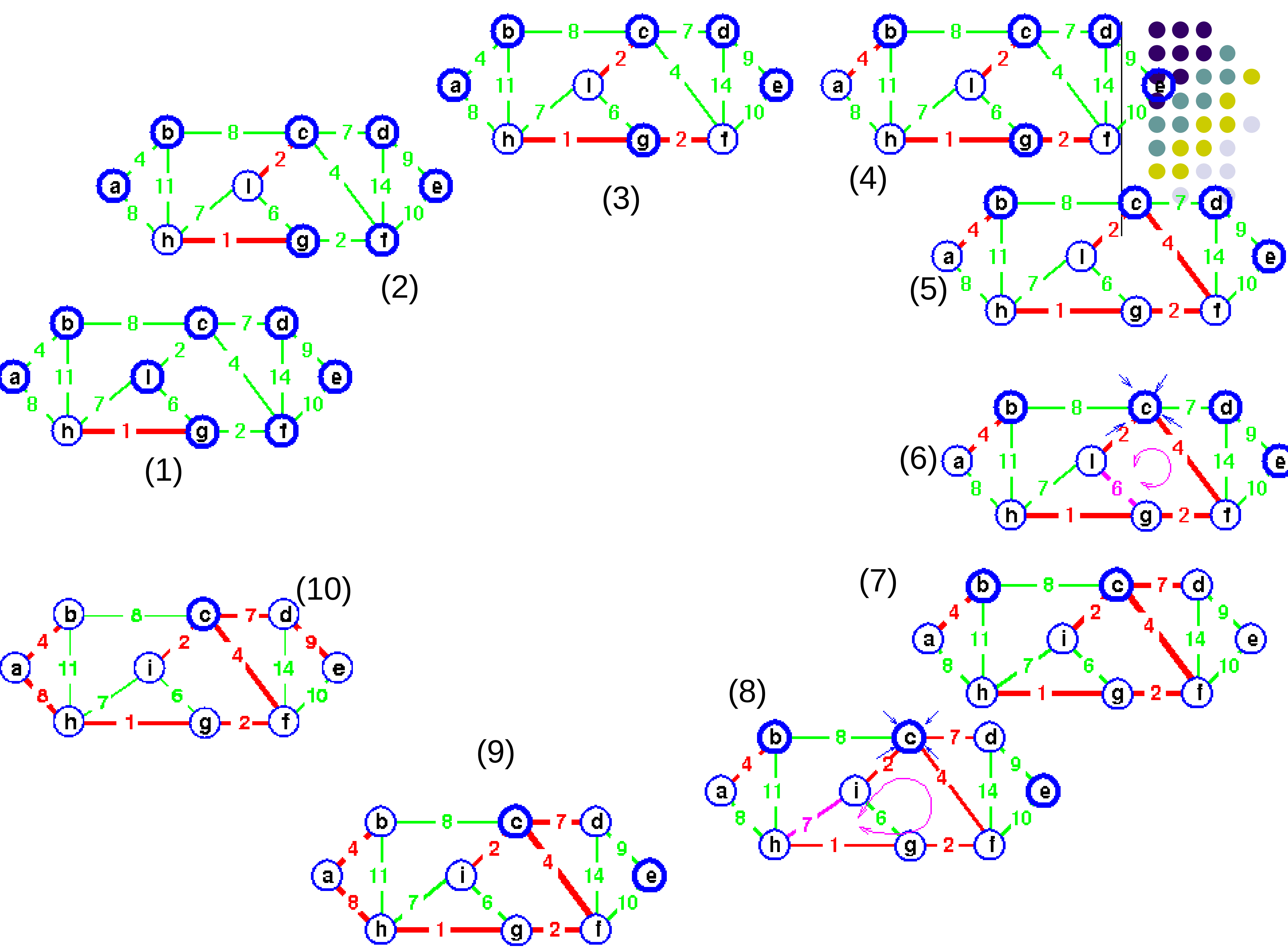
E(F) là tập cạnh của F.

$F = \varnothing$; $k = 0$;

while ($|E(F)| < n$-1)

1. Chọn cạnh $e \in E \setminus E(F)$ sao cho có trọng số nhỏ nhất, khi thêm vào E(F) không sinh ra chu trình;
2. $E(F) = E(F) \cup \{e\}$;

output F;


(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)
(10)

4. Kỹ thuật tham lam

<u>Nhận xét</u>:

Do F không có chu trình trong quá trình hình thành nên F là rừng.
Với cạnh e =(u,v) $\in$ E\E(F) có 4 khả năng:

a) u, v thuộc cũng một cây đã có trong F,

b) u, v thuộc 2 cây riêng biệt trong F,

c) u thuộc một cây đã có trong F và v ở ngoài F

d) cả u, v đều ở ngoài F

Cả 3 trường hợp b, c, d khi thêm vào F không sinh ra chu trình.

Đặt F = {V, E(F)=∅} - rừng gồm các cây ban đầu là các đỉnh của đồ thị G:

b), c), d) ⇔ *e* thuộc 2 cây riêng biệt

4. Kỹ thuật tham lam

Thiết kế chi tiết 1:

define Edge K;

1) sort on *t*: G[1].t ≤ G[2].t ≤ ....≤ G[m].t;
2) for (d = 1; d ≤ *n*; d++) cây(d) = d;
3) i = 0; j = 1;
4) **while** (i < *n*-1)
    a) while (G[j].d và G[j].c thuộc cùng 1 cây) j++;
    b) i ++; K(i) = G[j];
    c) p = số hiệu cây(G[j].d); q = số hiệu cây(G[j].c);
    d) Chập cây q vào cây p;
    e) j++;
5) **output** {K[1], ..., K[*n*-1]};

4. Kỹ thuật tham lam

Minh họa:

Cho đồ thị G={V,E},

V={1, 2, 3, 4, 5, 6};

E={(1,4,4), (3,5,5), (4,5,6), (4,3,7), (6,2,8), (1,5,9), (1,6,10), (2,5,11), (1,2,12)}

Mảng K	V	**1**	**2**	**3**	**4**	**5**	**6**
	Cây	*1*	*2*	*3*	*4*	*5*	*6*

1) Chọn cạnh K[1] = G[1], chập cây 4 vào 1

Mảng K	V	**1**	**2**	**3**	**4**	**5**	**6**
	Cây	*1*	*2*	*3*	*4*	*5*	*6*
		1	*2*	*3*	*1*	*5*	*6*

4. Kỹ thuật tham lam

Minh họa:
Cho đồ thị G={V,E},
V={1, 2, 3, 4, 5, 6};
E={(1,4,4), (3,5,5), (4,5,6), (4,3,7), (6,2,8), (1,5,9), (1,6,10), (2,5,11), (1,2,12)}

Mảng K	V	**1**	**2**	**3**	**4**	**5**	**6**
	Cây	*1*	*2*	*3*	*4*	*5*	*6*
		1	*2*	*3*	*1*	*5*	*6*

2) Chọn cạnh K[2] = G[2], chập cây 5 vào 3

Mảng K	V	**1**	**2**	**3**	**4**	**5**	**6**
	Cây	*1*	*2*	*3*	*4*	*5*	*6*
		1	*2*	*3*	*1*	*5*	*6*
		1	*2*	*3*	*1*	*3*	*6*

4. Kỹ thuật tham lam

Minh họa:

Cho đồ thị G={V,E},

V={1, 2, 3, 4, 5, 6}; E={(1,4,4), (3,5,5), (4,5,6), (4,3,7), (6,2,8), (1,5,9), (1,6,10), (2,5,11), (1,2,12)}

Mảng K	V	**1**	**2**	**3**	**4**	**5**	**6**
	Cây	*1*	*2*	*3*	*4*	*5*	*6*
		1	*2*	*3*	*1*	*5*	*6*
		1	*2*	*3*	*1*	*3*	*6*

3) Chọn cạnh K[3] = G[3], chập cây 3 vào 1

Mảng K	V	**1**	**2**	**3**	**4**	**5**	**6**
	Cây	*1*	*2*	*3*	*4*	*5*	*6*
		1	*2*	*3*	*1*	*5*	*6*
		1	*2*	*3*	*1*	*3*	*6*
		1	*2*	*1*	*1*	1	*6*

4. Kỹ thuật tham lam

Mảng K	V	**1**	**2**	**3**	**4**	**5**	**6**
	Cây	*1*	*2*	*3*	*4*	*5*	*6*
		1	*2*	*3*	*1*	*5*	*6*
		1	*2*	*3*	*1*	*3*	*6*
		1	*2*	*1*	*1*	1	*6*

4) Loại G[4], chọn cạnh K[4] = G[5], chập cây 6 vào 2

Mảng K	V	**1**	**2**	**3**	**4**	**5**	**6**
	Cây	*1*	*2*	*3*	*4*	*5*	*6*
		1	*2*	*3*	*1*	*5*	*6*
		1	*2*	*3*	*1*	*3*	*6*
		1	*2*	*1*	*1*	1	*6*
		1	*2*	*1*	*1*	1	2

4. Kỹ thuật tham lam

Mảng K	V	**1**	**2**	**3**	**4**	**5**	**6**
	Cây	*1*	*2*	*3*	*4*	*5*	*6*
		1	*2*	*3*	*1*	*5*	*6*
		1	*2*	*3*	*1*	*3*	*6*
		1	*2*	*1*	*1*	1	*6*
		1	*2*	*1*	*1*	1	2

5) Loại G[6], chọn cạnh K[5] = G[7], chập cây 2 vào 1

Mảng K	V	**1**	**2**	**3**	**4**	**5**	**6**
	Cây	*1*	*2*	*3*	*4*	*5*	*6*
		1	*2*	*3*	*1*	*5*	*6*
		1	*2*	*3*	*1*	*3*	*6*
		1	*2*	*1*	*1*	1	*6*
		1	*2*	*1*	*1*	1	2
		1	1	*1*	*1*	1	1

4. Kỹ thuật tham lam

Mảng K	V	**1**	**2**	**3**	**4**	**5**	**6**
	Cây	*1*	*2*	*3*	*4*	*5*	*6*
		1	*2*	*3*	*1*	*5*	*6*
		1	*2*	*3*	*1*	*3*	*6*
		1	*2*	*1*	*1*	1	*6*
		1	*2*	*1*	*1*	1	2
		1	1	*1*	*1*	1	1

Cây khung F={V,K},

K={G[1], G[2], G[3], G[5], G[7]},

μ(F) = 33

4. Kỹ thuật tham lam

<u>Thiết kế chi tiết 2</u>:

define Edge K;

1) sort on *t*: G[1].t ≤ G[2].t ≤ ....≤ G[m].t;
2) for (d = 1; d ≤ *n*; d++) sh(d) = d;
3) i = 0; j = 1;
4) **while** (i < *n*-1)
   a) while (sh[G[j].d] == sh[G[j].c] j++;
   b) i ++; K[i] = G[j];
   c) p = sh[G[j].d]; q = sh[(G[j].c];
   d) for (d =1;d ≤ n; d++)
      if (sh[d] = q) sh[d] = p;
   e) j++;
5) **output** {K[1], ..., K[*n*-1]};

4. Kỹ thuật tham lam

Nhận xét:

Thuật toán tham lam vốn chỉ cho lời giải gần đúng. Trong một số trường hợp, thuật toán có thể cho lời giải đúng.

Thuật toán Kruskal cho lời giải đúng.

4. Kỹ thuật tham lam

1. Mảng K chứa cây khung F:

   Thuật toán dừng và cho đủ *n*-1 cạnh.

   Thực vậy, nếu $j>m$ mà $k<n-1$, do khởi tạo có n cây, suy ra

   - Trong K có ít nhất 2 cây.
   - Thêm vào bất kỳ cạnh nào từ E\E(F) sinh ra chu trình.
   - Không có cạnh nào nối 2 cây.

   <u>Nghĩa là</u>: G không liên thông, mâu thuẫn với giả thiết G liên thông.

   Vậy, thuật toán dừng, F có *n*-1 cạnh, không có chu trình.

   Vậy, F là cây khung (liên thông, chứa cả *n* đỉnh).

4. Kỹ thuật tham lam

<u>2. F là MST của G</u>

Giả sử T* là MST có số cạnh chung với F nhiều nhất và $F \neq T^*$ (1)

Kí hiệu $e \in E(F)\backslash E(T^*)$:

$$f(e) = \min \{ f(x): x \in E(F)\backslash E(T^*) \}$$

$E(T^*) \cup \{e\}$ chứa chu trình, kí hiệu là C. Do F không chứa C nên chọn được

$$e1 \in E(C)\backslash E(F), \quad e1 \notin E(F)\cap E(T^*)$$

Suy ra

$$f(e1) \geq f(e)$$

(do e được chọn, e1 khôngđược chọn, dù khi thêm được vào $E(F)\cap E(T^*)$).

Xét đồ thị F1 = {E(F1), V}, với $E(F1) = E(T^*)\cup\{e\}\backslash\{e1\}$.

F1 là cây khung, $\mu(F1) \leq \mu(T^*)$, nghĩa là F1 là MST với

$$|E(F)\cap E(F1)| > |E(F)\cap E(T^*)|$$

Vậy F1 là MST có số lượng cạnh chung với F nhiều hơn cạnh chung của T* với F. Mâu thuẫn.

Nói cách khác, giả sử T* là MST có số cạnh chung với F nhiều nhất và $F \neq T^*$ là sai.

Suy ra: **T* = F**

4. Kỹ thuật tham lam

Thuật toán Kruskal:

input $\quad$ **V={1, 2, ..., n }, $\quad$ G[1..m] of egde**

output $\quad$ **V={1, 2, ..., n }, $\quad$ K[1..n-1] of egde**

Ý tưởng:

Kí hiệu $\quad$ **F là rừng sinh ra khi thực hiện thuật toán,**

E(F) là tập cạnh của F.

F = ∅; *k* = 0;

while (|E(F)| < *n*-1)

1. **Chọn cạnh e ∈ E\ E(F) sao cho có trọng số nhỏ nhất, khi thêm vào E(F) không sinh ra chu trình;**
2. **E(F) = E(F) ∪ {e};**

output F;

4. Kỹ thuật tham lam

<u>Thuật toán PRIM</u>

Kí hiệu F là đồ thị sinh ra trong thuật toán,

E(F) là tập cạnh và

V(F) là tập đỉnh của F.

1) k = 1;

2) E(F) = {e=(u,v): f(e)=min}; V(F) ={u,v};

3) while (|E(F)| < n-1)

Chọn cạnh e $\in$ E\ E(F) sao cho liên thuộc với V(F),

khi thêm vào E(F) mà không sinh ra chu trình và

có trọng số nhỏ nhất;

E(F) = E(F) $\cup$ {e};

4) output F;

4. Kỹ thuật tham lam

<u>Thuật toán PRIM</u>

1) k = 1;
2) E(F) = {e=(u,v): f(e)=min}; V(F) ={u,v};
3) **while** (|E(F)| < *n*-1)

   Chọn cạnh e $\in$ E\ E(F) sao cho liên thuộc với V(F),

   khi thêm vào E(F) mà không sinh ra chu trình và

   có trọng số nhỏ nhất;

   E(F) = E(F) $\cup$ {e};
4) **output** F;

<u>Phân tích các điều kiện để chọn cạnh e</u>:

F luôn là một cây nên để đảm bảo tính liên thuộc với V(F) và không sinh ra chu trình cạnh e=(u,v) phải thỏa mãn u, v không cùng thuộc V(F) và V\V(F).

Nếu kí hiệu VF là hàm thuộc đối với tập V(F) thì điều kiện chọn là VF(u)$\neq$VF(v).

Sắp xếp G[1].t ≤ ... ≤ G[m].t

cạnh e=(u,v) đầu tiên thỏa mãn VF(u)$\neq$VF(v) là cạnh chọn được.

4. Kỹ thuật tham lam

Thuật toán PRIM

1) k = 1;
2) E(F) = {e=(u,v): f(e)=min}; V(F) ={u,v};
3) **while** (|E(F)| < *n*-1)

   Chọn cạnh e $\in$ E\ E(F) sao cho liên thuộc với V(F),

   khi thêm vào E(F) mà không sinh ra chu trình và

   có trọng số nhỏ nhất;

   E(F) = E(F) $\cup$ {e};
4) **output** F;

Minh họa:

Cho đồ thị G={V,E},

V={1, 2, 3, 4, 5, 6};

E={(1,4,4), (3,5,5), (4,5,6), (4,3,7), (6,2,8), (1,5,9), (1,6,10), (2,5,11), (1,2,12)}

4. Kỹ thuật tham lam

```
1) sort on G[1].t ≤ ... ≤ G[m].t;
2) for ( i =1; i ≤ n; i++)   VF(i) =0;
3) i = 1;
4) P[i] = G[1];   VF(u) = 1;   VF(v) = 1;
5) while (i < n-1)
        a) j = 2;   while (VF(G[j].d) = VF(G[j].c))   j++;
        b) i++;   P[i] = G[j];
        c) VF(G[j].d) = 1;   VF(G[j].c) = 1;
6) output F;
```

4. Kỹ thuật tham lam

1) k = 1; E(F) = {e=(u,v): f(e)-min}; V(F) ={u,v};

2) while (|E(F)| < *n*-1)

Chọn cạnh e ∈ E\ E(F) sao cho liên thuộc với V(F), khi thêm vào E(F) mà không sinh ra chu trình và có trọng số nhỏ nhất;

E(F) = E(F) ∪ {e};

3) output F;

E={(1,4,4), (3,5,5), (4,5,6), (4,3,7), (6,2,8), (1,5,9), (1,6,10), (2,5,11), (1,2,12)}

Mảng P	VF	**1**	**2**	**3**	**4**	**5**	**6**
	i	0	0	0	0	0	0
G[1]	1	1	0	0	1	0	0
G[1],G[3]	2	1	0	0	1	1	0
G[1],G[3],G[2]	3	1	0	1	1	1	0
G[1],G[3],G[2],G[7]	4	1	0	1	1	1	1
G[1],G[3],G[2],G[7],G[5]	5	1	1	1	1	1	1

4. Kỹ thuật tham lam

<u>Ví dụ 4.2. </u>Thuật toán Dijkstra

Cho đồ thị liên thông G={V,E} và 2 đỉnh u, v $\in$V.

Tìm đường đi ngắn nhất α*(u,v).

4. Kỹ thuật tham lam

Ví dụ 4.2. Thuật toán Dijkstra

Cho đồ thị liên thông G={V,E} và 2 đỉnh u, v $\in$V.

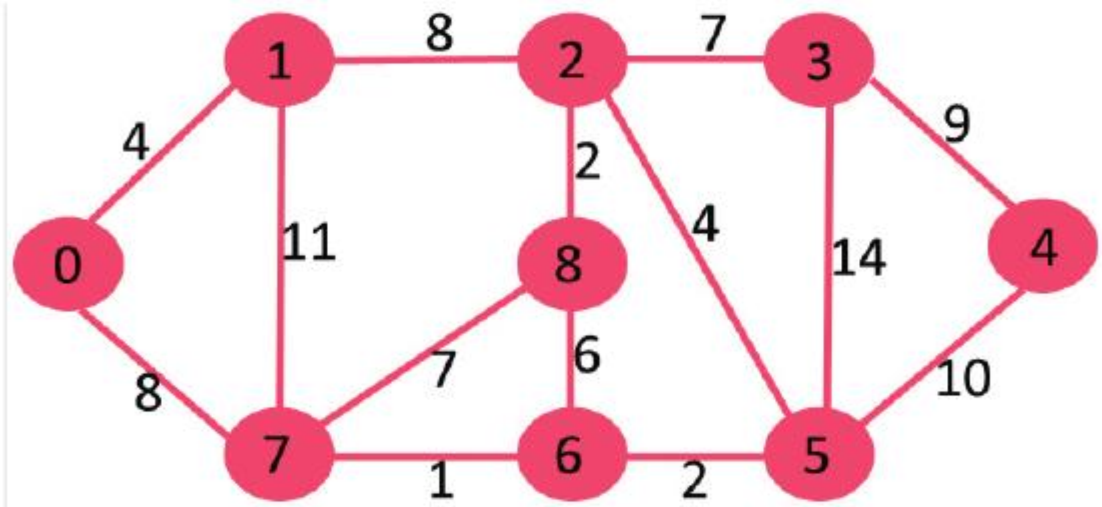
Tìm đường đi ngắn nhất α*(u,v).

Ý tưởng:

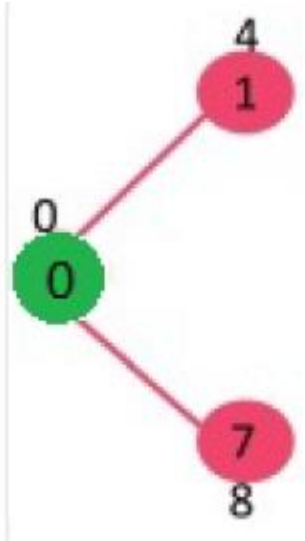
Từ đỉnh *u* loang dần ra xung quanh, cho đến khi gặp được đỉnh *v*.

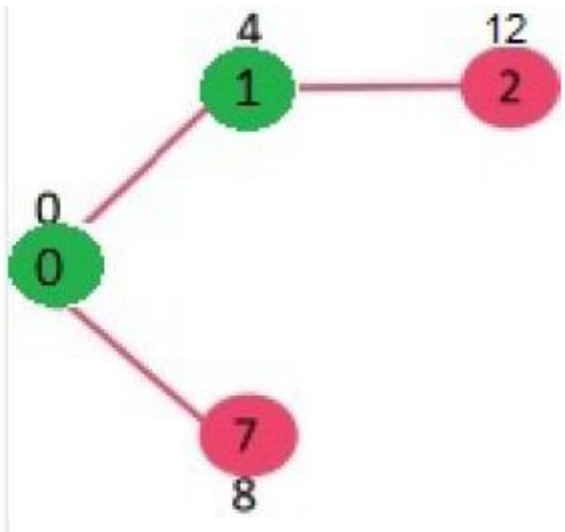
Mỗi bước sẽ xác định nhãn L cho các đỉnh nhằm đánh dấu khoảng cách (ngắn nhất có thể) đối với *u*.

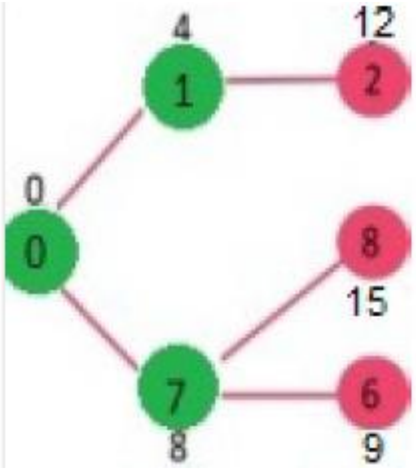
Khi *v* mang nhãn, thì đó cũng chính là khoảng cách (ngắn nhất) từ *u* đến *v*.



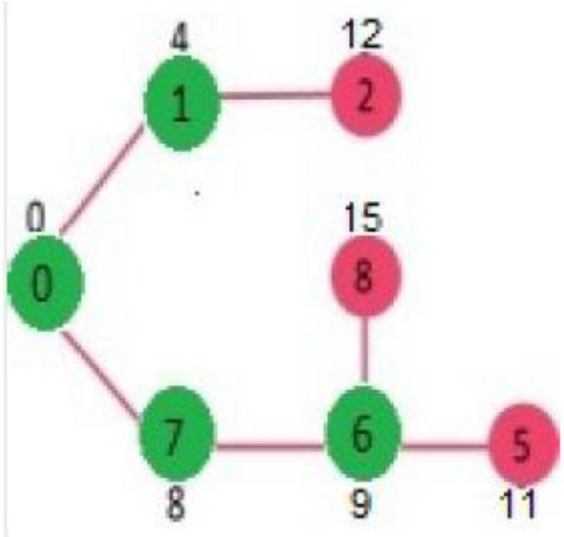
0
1
2
3
4
5
6
7
8
4
8
11
8
7
2
4
9
14
7
6
10
1
2


4
1
0
0
7
8

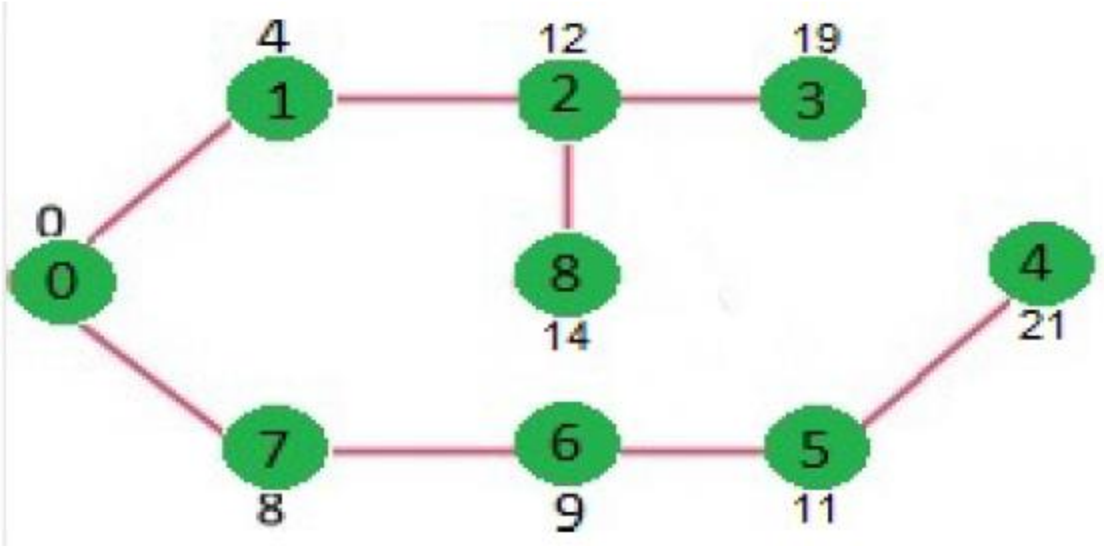

4
12
1
2
0
0
7
8



4
12
1
2
0
0
8
15
7
6
8
9



4
12
1
2
0
15
0
8
7
6
5
8
9
11


4
12
19
1
2
3
0
0
8
4
14
21
7
6
5
8
9
11

4. Kỹ thuật tham lam

Chiến lược loang:

Xét x và y thuộc danh sách kề (sau) của x.

Giả thiết ta đã biết độ dài đường đi (ngắn nhất) từ u tới x,

$$L(x) = \alpha^*(u, x)$$

và một độ dài đường đi nào đó tới y là $L(y)$ - nhãn của y. Nếu

$$L(y) > L(x) + f(x, y)$$

thì rõ ràng là có thể chọn đường đi mới từ u tới y là

$$\alpha(u, y) = \alpha^*(u, x) + f(x, y)$$

có độ dài L tốt hơn (ngắn hơn)

$$L(y) = L(x) + f(x, y)$$

4. Kỹ thuật tham lam

<u>Thiết kế 1</u>:

Kí hiệu S là tập các đỉnh có nhãn tốt.

1) **for (*x* = 1; *x* ≤ *n*; *x*++) {S(*x*) = 0; L(*x*) = +∞} (chưa đỉnh nào có nhãn tốt)**
2) **L(*u*) = 0;**
3) **while (S(*v*) = 0)**
   a) **Tìm *a*: L(*a*) = min {L(*x*): *x*∈V, S(*x*) = 0};**
   b) **S(*a*) = 1;**
   c) **for all *x* ∈V+(*a*)**
      if (L(*x*) > L(*a*) + *f*(*a*, *x*))
         L(*x*) = L(*a*) + *f*(*a*, *x*);

4. Kỹ thuật tham lam

<u>**Minh họa**</u>**:**

Xét đồ thị có hướng G = {V,E},

V = {1, 2, 3, 4, 5, 6},

E = { (1,5,2), (1,2,7), (2,6,4), (6,4,6), (3,4,2), (6,3,3),

(5,6,6), (5,2,1), (5,3,10)

}

Tìm α*(1,4).

4. Kỹ thuật tham lam

E = { (1,5,2), (1,2,7), (2,6,4), (6,4,6), (3,4,2), (6,3,3), (5,6,6), (5,2,1), (5,3,10) }.

Tìm $\alpha^*(1,4)$.

Bước 1:

Lặp	S		1	2	3	4	5	6
0		L	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1	1,		0	7	$+\infty$	$+\infty$	2	$+\infty$

Bước 2:

Lặp	**S**		**1**	**2**	**3**	**4**	**5**	**6**
0		L	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1	1,		0	7	$+\infty$	$+\infty$	2	$+\infty$
	1,5		0	3	12	$+\infty$	2	8

4. Kỹ thuật tham lam

E = { (1,5,2), (1,2,7), (2,6,4), (6,4,6), (3,4,2), (6,3,3), (5,6,6), (5,2,1), (5,3,10) }.

Tìm $\alpha^*(1,4)$.

Bước 2:

Lặp	**S**		**1**	**2**	**3**	**4**	**5**	**6**
0		L	0	+∞	+∞	+∞	+∞	+∞
1	1,		0	7	+∞	+∞	2	+∞
	1,5		0	3	12	+∞	2	8

Bước 3:

Lặp	**S**		**1**	**2**	**3**	**4**	**5**	**6**
0		L	0	+∞	+∞	+∞	+∞	+∞
1	1,		0	7	+∞	+∞	2	+∞
2	1, 5		0	3	12	+∞	2	8
3	1, 5, 2		0	3	12	+∞	2	7

4. Kỹ thuật tham lam

E = { (1,5,2), (1,2,7), (2,6,4), (6,4,6), (3,4,2), (6,3,3), (5,6,6), (5,2,1), (5,3,10) }.

Tìm $\alpha^*(1,4)$.

Bước 3:

Lặp	S		1	2	3	4	5	6
0		L	0	+∞	+∞	+∞	+∞	+∞
1	1,		0	7	+∞	+∞	2	+∞
2	1, 5		0	3	12	+∞	2	8
3	1, 5, 2		0	3	12	+∞	2	7

Bước 4:

Lặp	S		1	2	3	4	5	6
0		L	0	+∞	+∞	+∞	+∞	+∞
1	1,		0	7	+∞	+∞	2	+∞
2	1, 5		0	3	12	+∞	2	8
3	1, 5, 2		0	3	12	+∞	2	7
4	1, 5, 2, 6		0	3	10	13	2	7

4. Kỹ thuật tham lam

E = { (1,5,2), (1,2,7), (2,6,4), (6,4,6), (3,4,2), (6,3,3), (5,6,6), (5,2,1), (5,3,10) }. Tìm α*(1,4).

Bước 4:

Lặp	S		1	2	3	4	5	6
0		L	0	+∞	+∞	+∞	+∞	+∞
1	1,		0	7	+∞	+∞	2	+∞
2	1, 5		0	3	12	+∞	2	8
3	1, 5, 2		0	3	12	+∞	2	7
4	1, 5, 2, 6		0	3	10	13	2	7

Bước 5:

Lặp	S		1	2	3	4	5	6
0		L	0	+∞	+∞	+∞	+∞	+∞
1	1,		0	7	+∞	+∞	2	+∞
2	1, 5		0	3	12	+∞	2	8
3	1, 5, 2		0	3	12	+∞	2	7
4	1, 5, 2, 6		0	3	10	13	2	7
5	1, 5, 2, 6, 3		0	3	10	12	2	7

4. Kỹ thuật tham lam

E = { (1,5,2), (1,2,7), (2,6,4), (6,4,6), (3,4,2), (6,3,3), (5,6,6), (5,2,1), (5,3,10) }. Tìm α*(1,4).

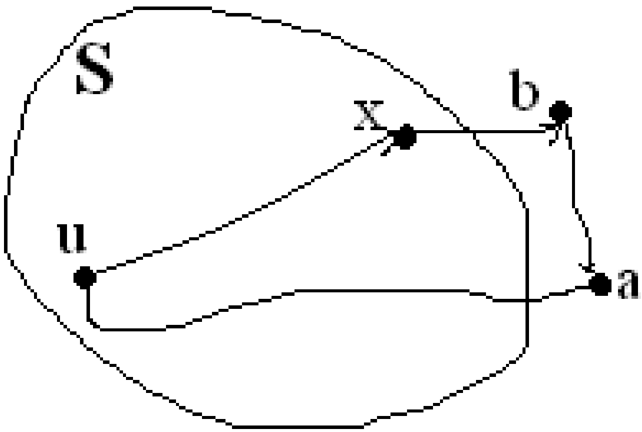
Bước 5:

Lặp	S		1	2	3	4	5	6
0		L	0	+∞	+∞	+∞	+∞	+∞
1	1,		0	7	+∞	+∞	2	+∞
2	1, 5		0	3	12	+∞	2	8
3	1, 5, 2		0	3	12	+∞	2	7
4	1, 5, 2, 6		0	3	10	13	2	7
5	1, 5, 2, 6, 3		0	3	10	12	2	7

Bước 6:

Lặp	S		1	2	3	4	5	6
0		L	0	+∞	+∞	+∞	+∞	+∞
1	1,		0	7	+∞	+∞	2	+∞
2	1, 5		0	3	12	+∞	2	8
3	1, 5, 2		0	3	12	+∞	2	7
4	1, 5, 2, 6		0	3	10	13	2	7
5	1, 5, 2, 6, 3		0	3	10	**12**	2	7
6	1, 5, 2, 6, 3, 4. **Stop**							

4. Kỹ thuật tham lam



<u>Chứng minh tính đúng đắn của thuật toán</u>:

L(*a*) ≤ L(*y*), $\forall$ *y* $\in$ V\S;

$\mu(\alpha^*(u,a)) < L(a)$

Xét cạnh (x,b) đầu tiên ra khỏi S trên $\alpha^*(u,a)$, b $\notin$ S:

có L(a) ≤ L(b),

Mặt khác, L(a) $> \mu(\alpha^*(u,a))$

$\geq \mu(\alpha^*(u,x)) + f(x,b)$

$= L(x) + f(x,b)$

$= L(b)$

Vô lý!□

4. Kỹ thuật tham lam

Xác định các đỉnh trên đường đi $\alpha^*(u, v)$:

4) p(1) = v; i = 1;

5) while (p(i)$\neq u$)

    a) Tìm $x \in V^{-}(p(i))$: $L(x) + f(x, p(i)) = L(p(i))$;

    b) p(i+1) = x;

    c) i++;

6) output {p(i), ..., p(1)};

<u>Bài tập (*tự làm*)</u>:

Theo các bước 4, 5, 6 tìm dãy các đỉnh trên đường đi có độ dài 12 với dữ liệu có trong bảng.

4. Kỹ thuật tham lam

Ví dụ 4.3. Thuật toán tô màu đồ thị.

Tô màu đồ thị là gán cho mỗi đỉnh một số hiệu màu sao cho hai đỉnh liền kề thì không cùng số màu. Tìm phương án tô màu sao cho số màu phải dùng là ít nhất.

Ý tưởng:

1) số hiệu màu được dùng m = 1;
2) xét từng đỉnh $v \in V$
   a) xét các màu i $\in \{1, ..., m\}$
      nếu tô màu cho v bởi màu i được thì đặt shm[v] = i;
   b) nếu v không được tô màu
      m = m +1;
      shm[v] = m;

4. Kỹ thuật tham lam

1) số hiệu màu được dùng m = 1;
2) xét từng đỉnh v ∈ V
   a) xét các màu i ∈ { 1, ..., m}
      nếu tô màu cho v bởi màu i được thì đặt shm[v] = i;
   b) nếu v không được tô màu
      m = m +1;
      shm[v] = m;

Trong chiến lược loang 2) có thể đưa thêm thông số bậc: đỉnh nào có số bậc cao hơn (liên quan đến nhiều đỉnh khác hơn) thì xét trước.

4. Kỹ thuật tham lam

Minh họa:

Đỉnh	Danh sách kề	màu
a	b, d, c	1
b	a, v, c	2
c	u, v, a, b, d	3
u	d, c, v	1
v	u, c, b	4
d	a, c, u	2

4. Kỹ thuật tham lam

Ví dụ 4.4. Giải bài toán cái túi

Cho *n* đồ vật có kích thước a_1, a_2, ..., a_n, với $a_i \le T$, $i = 1, 2, ..., n$.

Cần sử dụng ít nhất bao nhiêu túi để chứa hết các đồ vật.

<u>Ý tưởng</u>:

xét từ đồ vật thứ i, i = 1, 2, ..., n

túi j = 1;

while (xếp đồ vật thứ i vào túi j không vừa) j++

xếp đồ vật j vào túi i;

Minh họa 10 đồ vật: 0.2, 0.3, 0.4, 0.5, 0.4, 0.2, 0.3, 0.2, 0.3, 0.2

túi: T = 1

túi 1: 1, 2, 3 (0.2, 0.3, 0.4)

túi 2: 4, 5 (0.5, 0.4)

túi 3: 6, 7, 8, 9 (0.2, 0.3, 0.2, 0.3)

túi 4: 10 (0.2)

4. Kỹ thuật tham lam

<u>Câu hỏi ôn tập và bài tập tự làm</u>:

1. Khi nào thuật toán được xây dựng trên kỹ thuật tham lam cho lời giải chính xác?
2. Cho *n* đồ vật có trọng lượng a1, a2, ..., an và giá trị tương ứng là c1, c2,.., cn. Xây dựng thuật toán xếp đồ vật vào túi có sức chứa T sao cho tổng giá trị các đồ vật được xếp vào túi là lớn nhất.
3. Người ta may sẵn *n* cái áo với các kích thước vai V(1), V(2), …, V(n) cho *n* học sinh. Các học sinh này được đánh số từ 1 tới *n* và có kích thước vai là p(1), p(2), …, p(*n*). Nếu học sinh *i* được nhận áo *j* thì độ lệch vai cho trường hợp này sẽ là |V(*i*)-p(*j*)|. Một phương án phân phối áo cho các học sinh 1, 2, ...*n* được mô tả như một hoán vị π1, π2, ..., π*n* với hàm ý học sinh *i* nhận được áo π*i*. Độ lệch của phương án này được định nghĩa bằng

   $$\max \{|V(i) - p(\pi i)|, i=1,2,...,n\}$$

   Hãy xây dựng thuật toán phân phối áo cho các học sinh sao cho độ lệch là cực tiểu.
4. Có *n* cuộc họp, cuộc họp thứ *i* bắt đầu vào thời điểm *a*i và kết thúc ở thời điểm *b*i. Hai cuộc họp được bố trí tại cùng một phòng nếu khoảng thời gian làm việc của chúng giao nhau không nhiều hơn thời điểm tiếp nối. Hãy bố trí phòng họp sao cho phải dúng ít phòng nhất.
5. Tìm hiểu thuật toán leo đồi đối với bài toán tối ưu.