KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
NGÀNH CÔNG NGHỆ THÔNG TIN

BÁO CÁO CUỐI KỲ

NĂM HỌC 2021-2022

Giáo viên giảng dạy : Nguyễn Quang Ngọc
STT: 10 Tên sinh viên : Nguyễn Thành Đạt MSSV : 20110121

A. KIẾN THỨC CƠ BẢN :

I. Công thức đa thức tối tiểu :

1. Định nghĩa :

   Công thức đơn giản nhất được gọi là công thức đa thức tối tiểu

2. Phương pháp : PHƯƠNGNG PHÁP THỎA THUẬN

- Phương pháp :

  Bước 1: Viết f dưới dạng tổng các tích (nối rời chính tắc).

  $f = m_1 \vee m_2 \vee . . \vee m_r$

  Đặt $T = \{m_1 , m_2 , . . , m_r \}$

  Gọi V là danh sách các biến theo một thứ tự nhất định nhưng tùy ý.

  Bước 2: Cập nhật T.

i) Giả sử x là một biến trong V mà ta đang xét tới.

   T đã được cập nhật theo các biến trước x.

   Phân hoạch T thành :

   * A : gồm các đơn thức có biến x (đơn thức chia hết cho x).

   * B : gồm các đơn thức có biến -x.

ii) Nếu A = ∅ hay B = ∅ thì xét biến kế x trong V.

   Nếu không, thêm vào T tất cả các thỏa thuận của một đơn thức thuộc A và một đơn thức thuộc B.

ii) Mỗi lần thêm một phần tử mới vào T , ta phải loại bớt những phần tử được trội thực sự bởi một phần tử khác (kể cả phần tử mới thêm vào).

Bước 3: Giả sử khi đã duyệt hết các biến ta có

$$T = \{ M_1 , M_2 , \ldots , M_p \}$$

i) Với mỗi từ tối tiểu ti trong cách biểu diễn f dưới dạng nối rời chính tắc, gọi $N_i = \{ M_{i1}, M_{i2}, \ldots , M_{ik} \} \subseteq T$, M_{ij} trội t_i .

ii) Loại bỏ sự trùng lắp: Nếu $N_i \subseteq N_j$ thì loại N_j .

iii) Một công thức của f là tổng của bộ $(a_1 , a_2 , \ldots , a_q)$ với $a_i \in N_i$ (còn lại).

iv) So sánh các công thức.

- Bài toán : Tìm công thức đa thức tối tiểu của hàm Bool 4 biến x, y, z, t
  f = $\bar{x}\bar{y}\bar{t} \vee \bar{x}\bar{z}t \vee x\bar{y}z\bar{t} \vee x\bar{y}\bar{z}\bar{t}$ .Thứ tự duyệt các biến x, y, z, t.

Bươc 1 :

T = {-x-y-t, -x-z-t, x-yz-t, x-y-z-t}, V= {x, y, z, t}

Bước 2 :

x : A = {x-yz-t, x-y-z-t} , B = {-x-y-t, -x-z-t, }
TT(thỏa thuận) = {-yz-t, -y-z-t,}, T = {-yz-t, -y-z-t, -x-y-t, -x-z-t}

y: A = {∅}, B = {-yz-t, -y-z-t, -x-y-t, -x-z-t}
TT(thỏa thuận) = {∅}, T = {-yz-t, -y-z-t, -x-y-t, -x-z-t}

z: A = {-yz-t}, B = {-y-z-t, -x-z-t}
TT(thỏa thuận) = {-y-t, -x-y-t}, T = {-y-t, -x-z-t}

t: A = {∅}, B = {-y-t, -x-z-t}
TT(thỏa thuận) = {∅}, T ={-y-t, -x-z-t}

Bước 3 :

T = {-x-y-t, -x-z-t, x-yz-t, x-y-z-t}
⇨ T ={-y-t, -x-z-t}
i) N1 = {-y-t}
N2 = {-x-z-t}
N3 = {-y-t}
N4 = {-y-t}
ii) N1 , N2

- Kết quả (chỉ có điểm khi các bước trên đúng) :
  f = -y-t V -x-z-t

II. Cây khung :

1. Định nghĩa : T là cây khung của một đồ thị G =(V, E) nếu T là một cây có tập đỉnh là tập đỉnh V và tập cạnh là tập con của E của G.

2. Phương pháp : THUẬT TOÁN BFS

- Thuật toán :

Bước 1:

i=0

Với mỗi u ∈ V thực hiện D[u]= 0 ; // u chưa được duyệt

D[s]=1; // s đã được duyệt

p[s]= -1;

Thêm đỉnh s vào Q;

Bước 2 :

i=1

while (Q≠∅)

{

1. u = Phần tử đỉnh Q; // Chỉ sao chép giá trị

2. Với mỗi v kề u thực hiện (Duyệt theo thứ tự)

3. if (D[v]=0)

{

4. D[v]= 1;

5. p[v] = u;

6. Thêm v vào Q ;

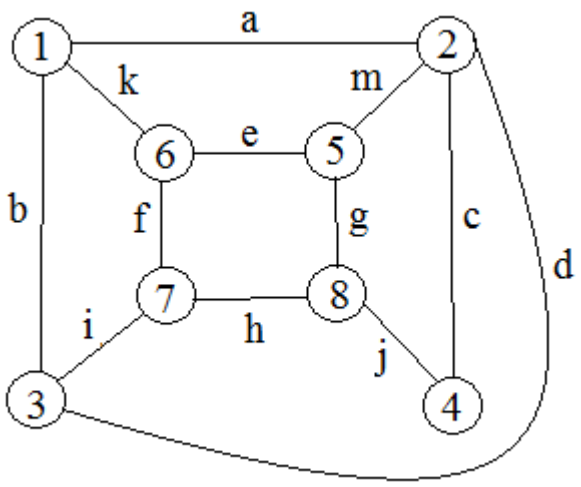
} // End If

7. Xóa phần tử đỉnh Q;

8. i=i+1;

} // End While

- Bài toán : Tìm cây khung của G sau :



- Gốc : đỉnh 4,
- Thứ tự duyệt : 1 2 3 4 5 6 7 8,
- **Bảng** (Chỉ có điểm khi có bảng):

i	**u**	**Q**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0		4	0,	0,	0,	1, -1	0,	0,	0,	0,
1	4	824		1,4						1,4
2	2	3182	1,2		1,2					
3	8	75318					1,8		1,8	
4	1	67531						1,1		
5	3	6753								
6	5	675								
7	7	67								
8	6	6								
9		Rỗng								

Các cạnh của cây khung kết quả (viết theo thứ tự alphabet): c, j, a, d, g, h, k

III. Đường đi ngắn nhất :

1. Định nghĩa :

Cho G=(V, E), có trọng số , có hướng.

- Trọng số của một đường đi từ đỉnh v đến w là tổng trọng số các cạnh của đường đi đó.

- Đường đi ngắn nhất giữa 2 đỉnh v, w là đường đi trọng số bé nhất.

2. Thuật toán : THUẬT TOÁN DIJKSTRA

- Thuật toán :

Bước 1 :

i = 0

Với mọi v ∈ V, d[v]=∞;

d[s]=0;

p[s]= - 1;

T=V;

Bước 2 :

i = 1 // Bước lặp thứ i.

While T ≠∅

{

1. Chọn u ∈T với d[u] bé nhất.

2. T=T-{u};

3. Với mỗi v ∈ T kề với u ((u,v) ∈ E) thực hiện (if . . .):

if (d[v] > d[u]+wuv)

{

3.1. d[v] = d[u]+wuv

3.2. p[v] = u

}

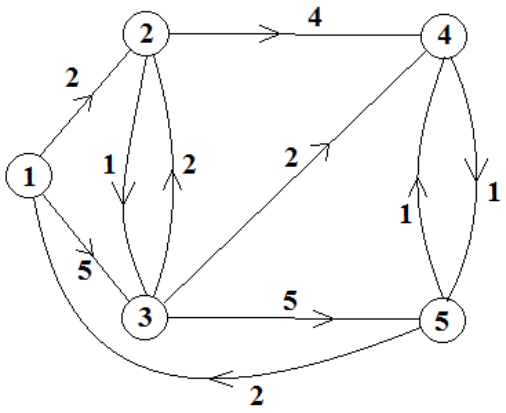
4. i = i + 1

} (Kết thúc While)

Bước 3 :

Viết d, p.

- Bài toán : Tìm đường đi ngắn nhất từ đỉnh 2 đến các đỉnh còn lại của đồ thị sau :



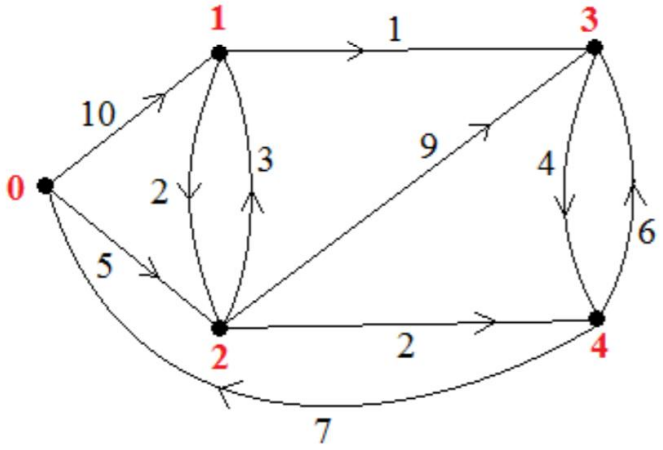
- Bảng :

i	u	d[1],p[1]	d[2],p[2]	d[3],p[3]	d[4],p[4]	d[5],p[5]
0		∞,	0, -1	∞,	∞,	∞,
1	2			1, 2	4, 2	
2	3				3, 3	6, 3
3	4					4, 4
4	5	6, 5				
5	1					
6		6, 5	0, -1	1, 2	3, 3	4, 4

B. CÀI ĐẶT CHƯƠNG TRÌNH CHO THUẬT TOÁN DIJKSTRA BẰNG NGÔN NGỮ C / C++ :

1. Giải thích file dữ liệu để đọc

Ví dụ:



Ta có ma trận trọng số tương đương của đồ thị trên:

```
5
0       10      5       -1      -1
-1      0       2       1       -1
-1      3       0       9       2
-1      -1      -1      0       4
7       -1      -1      6       0
```

- 5 là số đỉnh
- Định nghĩa: (i, j là các đỉnh của thị)

  + 0 nếu i = j; (các số 0 của ma trận trên)

  + W_{ij} = trọng số của cạnh có hướng (i,j) ∈ E (các số khác 0, -1 của ma trận trên)

  + -1 nếu i ≠ j và (i, j) ∉ E. (các số -1 của ma trận trên)

2. Chương trình cài đặt thuật toán Dijkstra

```
#include <stdio.h>

#define duongvocung 2147483647 // dinh nghia so duong vo cung
#define max 100 // dinh nghia so phan tu lon nhat cua mang
```

```
void XoaPhanTu(int A[max], int &n, int pt);
void thuattoan_Dijkstra(int A[max][max], int n, int dd, int dc);

int main()
{
	printf("Nguyen Thanh Dat ------ 20110121\n\n");
	int A[max][max];
	int n; // ma tran vuong cap n
	FILE *fp;  // con tro tap tin
	fp = fopen("D:\\DATATRR\\Dijkstra.txt", "r"); // duong dan
	if (fp == NULL)  // kiem tra file co doc duoc hay khong
	{
		printf("ERROR!");
		return -1;
	}
	fscanf(fp, "%d", &n);
	for( int i=0; i<n; i++) // doc phan tu trong ma tran
	{
		for( int j=0; j<n; j++)
		{
			fscanf(fp, "%d", &A[i][j]);
		}
	}
	fclose(fp);
	printf("Ma tran trong so: \n"); // in ma tran trong so tu file
	for( int x=0; x<n; x++)
	{
		for( int y=0; y<n; y++)
```

```
            {
                printf("%7d", A[x][y]);
            }
            printf("\n");
        }
        printf("\n");
        int dxp;
        printf("Nhap dinh xuat phat: ");
        scanf("%d", &dxp);
        while( (dxp<0) || (dxp>(n-1)) ) // kiem tra dinh xuat phat nhap vao tu
0 -> n-1
        {
            printf("Nhap lai dinh xuat phat: ");
            scanf("%d", &dxp);
        }
        int dkt;
        printf("Nhap dinh ket thuc: ");
        scanf("%d", &dkt);
        while( (dkt<0) || (dkt>(n-1)) ) // kiem tra dinh ket thuc nhap vao tu 0 -
> n-1
        {
            printf("Nhap lai dinh ket thuc: ");
            scanf("%d", &dkt);
        }
        printf("\n");
        thuattoan_Dijkstra(A, n, dxp, dkt);
        return 0;
}
```

```
void thuattoan_Dijkstra(int A[max][max], int n, int dd, int dc) // thuat toan dijkstra
{
	int d[max]; // mang d de chua tong trong so tren duong di
	int p[max]; // mang p de chua gia tri duong di
	int T[max]; // chua phan tu de duyet
	int u;
	// buoc 1:
	for( int v=0; v<n; v++)
	{
		d[v] = duongvocung;
	}
	d[dd] = 0;
	p[dd] = -1;
	for( int i=0; i<n; i++)
	{
		T[i] = i;
	}
	// buoc 2:
	int length = n; // so phan tu cua T
	while( length != 0 ) // kiem tra so phan tu cua T
	{
		int min = d[T[0]]; // Chon u thuoc T voi d[u] be nhat.
		u = T[0];
		for( int i1=0; i1<length; i1++)
		{
			if(d[T[i1]] < min)
```

```
                {
                        u = T[i1];
                }
        }
        XoaPhanTu(T, length, u); // thuc hien xoa phan tu da kiem tra
        for( int i2=0; i2<length; i2++)
        {
                int v = T[i2];
                if( A[u][v] > 0 ) // kiem tra v thuoc T co ke voi u khong? ((u,v) thuoc E)
                {
                        if( d[v] > d[u]+A[u][v]) // so sanh neu dung cap nhat lai d[v], p[v]
                        {
                                d[v] = d[u]+A[u][v];
                                p[v] = u;
                        }
                }
        }
} // ket thuc while
printf("Ket qua:\n"); //in ket qua cuoi cung
for( int kq=0; kq<n; kq++)
{
        printf("(%d, %d)   ", d[kq], p[kq]);
}
printf("\n");
printf("\n");
```

```
	if( (d[dc] == duongvocung) ) // kiem tra xem co duong di tu diem dau den diem cuoi ko
		printf("Khong co duong di tu %d -> %d!!!", dd, dc);
	else
		printf("Tong trong so tren duong di ngan nhat tu %d -> %d: %d\n\n", dd, dc, d[dc]);

	int B[10];
	B[0] = dc; // mang chua duong di
	int dem = 1;
	int x = p[dc];
	while(x != -1) // kiem tra duong di cho den diem dau
	{
		B[dem++] = x;
		x = p[x]; // di den diem tiep theo
	}
	printf("Duong di ngan nhat tu %d -> %d:\n", dd, dc);
	for( int i3=dem-1; i3>=0; i3--)
	{
		printf("%d -> ", B[i3]);
	}
	printf("END\n");
}

void XoaPhanTu(int A[max], int &n, int pt) // thuc hien xoa phan tu cua mang roi giam so luong phan tu
{
	int vt;
```

```
        for( int i=0; i<n; i++) // tim kiem vi tri cua phan tu can tim trong mang
        {
                if(A[i] == pt)
                {
                        vt = i;
                        break;
                }
        }
        for( int j=vt; j<n; j++) // thuc hien xoa phan tu
        {
                A[j] = A[j+1];
        }
        n--; // giam so luong phan tu
}
```