

Bài Toán Người Du lịch tham lam

```
#include<iostream>
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
#include<fstream>

using namespace std;
fstream f;
int n, a[100][100], TOUR[100], dem1=0, dem2=0, dem3=0;
void Init() {
    f.open("nguoidulich5.txt");
    f >> n;
    for (int i = 1; i <= n; i++)
        for (int j = 1; j <= n; j++) {
            f >> a[i][j];
        }
    f.close();
}
int GTS(int a[][100], int n, int TOUR[100], int dau)
{
    int v, k, w, mini, COST, daxet[100];
    for (int k = 1; k <= n; k++)
    {
        daxet[k] = 0;
    }
    COST = 0;
    int i;
    v = dau;
    i = 1;
    TOUR[i] = v;
    daxet[v] = 1;
    while (i < n)
    {
        mini = INT16_MAX;
        for (k = 1; k <= n; k++)
        {
            dem1++;
            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;
    }
    COST += a[v][dau];
    return COST;
}
```

```

}

void Result() {
    cout << "\n T1->";
    for (int i = 2; i <= n; i++) cout << "T" << TOUR[i] << "->";
    cout << "T1";
}

int main()
{
    Init();
    int kq = GTS(a, n, TOUR, 1);
    Result();
    cout << "Ket qua la: " << kq;
    system("pause");
}

```

Bài toán người du lịch theo phương pháp quay lui

```

//Giai theo phuong phap quay lui
#include<iostream>
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
using namespace std;
int n, c[100][100], x[100], chuaxet[100], kq[100];
int MIN = INT16_MAX;
void Init() {
    cout << "\n Nhap n="; cin >> n;
    cout << "\n Nhap chi phi \n";
    for (int i = 1; i <= n; i++)
        for (int j = 1; j <= n; j++) {
            if (i != j) {
                cout << "c[" << i << "][" << j << "]="; cin >> c[i][j];
            }
            else c[i][j] = 0;
        }
    x[1] = 1;
    for (int i = 2; i <= n; i++) chuaxet[i] = 1;
}
void Result() {
    cout << "\n T1->";
    for (int i = 2; i <= n; i++) cout << "T" << kq[i] << "->";
    cout << "T1";
    cout << "\n Tong chi phi la: " << MIN;
}
void Work() {
    int S = 0;
    for (int i = 1; i <= n - 1; i++) {
        S = S + c[x[i]][x[i + 1]];
    }
    S = S + c[x[n]][1];
    if (S < MIN) {
        MIN = S;
        for (int i = 1; i <= n; i++) kq[i] = x[i];
    }
}
}

```

```

void Try(int i) {
    for (int j = 2; j <= n; j++) {
        if (chuaxet[j]) {
            x[i] = j;
            chuaxet[j] = 0;
            if (i == n)
                Work();
            else
                Try(i + 1);
            chuaxet[j] = 1;
        }
    }
}
void main() {
    Init();
    Try(2);
    Result();
    system("pause");
}

```

Bài cái túi theo phương pháp chia để trị(slide 4+5)

```

#include<iostream>
#include<fstream>
using namespace std;
#define MAX 100

int n, m;
// n la so do vat va m la khoi luong toi da ma tui mang duoc
int X[MAX], Y[MAX];
// X[i], Y[i] de luu khoi luong va gia tri cua vat thu i
int F[MAX][MAX];
//F[i][j] la tong gia tri lon nhat ma tui chua duoc neu lay tu 1...i vat
// voi khoi luong gioi han la j

void Enter()
{
    fstream f;
    f.open("BAICAITUI.txt");
    f >> n >> m;
    for (int i = 1; i <= n; ++i)
        f >> X[i] >> Y[i];
}

void Init()
{
    for (int i = 1; i <= n; ++i)
        F[i][0] = 0;
}

void Optimize()
{
    for (int i = 1; i <= n; ++i)

```

```

        for (int j = 0; j <= m; ++j)
        {
            F[i][j] = F[i - 1][j];
            //Gia su chi chon duoc den vat thu i
            if (j >= X[i] && F[i][j] < F[i - 1][j - X[i]] + Y[i])
                F[i][j] = F[i - 1][j - X[i]] + Y[i];
        }
    }

void Trace() //Truy vet
{
    std::cout << "Gia tri Max = " << F[n][m];
    std::cout << std::endl;

    while (n != 0)
    {
        if (F[n][m] != F[n - 1][m]) // Co the chon duoc vat thu n
        {
            std::cout << n << std::endl;
            m -= X[n];
            // Da chon vat thu n thi chi con mang duoc m-Y[n] khoi luong
        }
        --n;
    }
}

void main()
{
    Enter();
    Init();
    Optimize();
    Trace();
    system("pause");
}

```

Bài toán Mergesort theo phương pháp chia để trị

```

// Bài Toán MergeSort:
/*
    Ý tưởng:
        Nếu có 2 dãy a và b đã được sắp xếp, tiến hành trộn 2 dãy này thành dãy c
        đã được sắp xếp
        Nếu chia nhỏ mảng cần sắp xếp thành các đoạn 1 phần tử thì mảng này đã đc
        sắp xếp
        Tiến hành ghép các đoạn nhỏ thành các đoạn lớn đã được sắp xếp
*/
#include<iostream>
using namespace std;

void xuat(int a[], int l, int r) {
    int i;
    for (i = l; i <= r; i++) {
        printf("%d \t", a[i]);
    }
}

```

```

    }
    printf("\n");
}

/* tron 2 mang
* mang a[l -> m] da sap xep
* mang a[m+1 -> r] da sap xep
* */
void tron(int a[], int l, int m, int r) {
    int i, length;
    int l_end = m - 1;
    i = l;
    length = r - l;

    int temp[100];

    /* tron cac phan tu cua 2 mang con cua a*/
    while (l <= l_end && m <= r) {
        if (a[l] <= a[m]) {
            temp[i++] = a[l++];
        }
        else {
            temp[i++] = a[m++];
        }
    }

    /* Neu mang dau tien chua het */
    while (l <= l_end) {
        temp[i++] = a[l++];
    }

    /* Neu mang thu 2 chua het*/
    while (m <= r) {
        temp[i++] = a[m++];
    }

    for (i = 0; i <= length; i++, r--) {
        a[r] = temp[r];
    }
}

/* thuat toan sap xep tron*/
void sx_tron(int a[], int l, int r) {
    //xuat(a, l, r);
    int m;
    if (l < r) {
        m = (l + r) / 2;
        sx_tron(a, l, m);
        sx_tron(a, m + 1, r);
        tron(a, l, m + 1, r);
    }
}

void main()
{
    int a[] = { 2, 6,875854, 3, 8, 5 ,3,5,7,2,7
,434,6565,344,878,34,543457,686,54353,};
    int n = 19;

```

```

        sx_tron(a, 0, n - 1);
        xuat(a, 0, n - 1);
        system("pause");
    }

```

Bài toán tìm kiếm nhị phân

```

#include<stdio.h>
#include<iostream>
using namespace std;
int TimKiemNhiPhan(int a[], int x, int left, int right)
{
    if (left > right)
        return 0;
    else
    {
        int mid = (left + right) / 2;
        if (x == a[mid])
            return 1;
        else
        {
            if (x > a[mid])
                TimKiemNhiPhan(a, x, mid + 1, right);
            else
            {
                TimKiemNhiPhan(a, x, left, mid - 1);
            }
        }
    }
}
void main()
{
    int a[] = { 3,5,12,16,19,34,45,47,51,57 };
    int tk = TimKiemNhiPhan(a, 40, 0, 9);
    if (tk == 1)
        printf("Tim thay!");
    else
    {
        printf("Khong tim thay");
    }
    system("pause");
}

```

Bài toán Tìm minmax bằng phương pháp chia để trị

```

#include<stdio.h>
#include<iostream>
using namespace std;

```

```

void MinMax(int a[], int l, int r, int &min, int &max)
{
    int min1, min2, max1, max2;
    if (l == r)
    {
        min = a[l];
        max = a[l];
    }
    else
    {
        MinMax(a, l, (l + r) / 2, min1, max1);
        MinMax(a, (l + r) / 2 + 1, r, min2, max2);
        if (min1 < min2)
        {
            min = min1;
        }
        else
        {
            min = min2;
        }
        if (max1 > max2)
        {
            max = max1;
        }
        else
            max = max2;
    }
}

void main()
{
    int a[] = { 5, 2, 6, 45, 3, 396, 66, 4 };
    int min, max;
    MinMax(a, 0, 7, min, max);
    cout << "Max= " << max;
    cout << "Min= " << min;
    system("pause");
}

```

Bài toán tìm dãy con theo phương pháp chia để trị

```

#include<stdio.h>
#include<iostream>

int max(int a, int b, int c)
{
    int m = a;
    if (b > m)
        m = b;
    if (c > m)
        m = c;
    return m;
}

int max(int a, int b)
{
    int m = a > b ? a : b;
}

```

```

        return m;
    }
    int maxsubleft(int a[], int i, int j)
    {
        int maxsum = -INT_MAX;
        int sum = 0;
        for (int k = j; k >= i; k--)
        {
            sum = sum + a[k];
            maxsum = max(sum, maxsum);
        }
        return maxsum;
    }
    int maxsubright(int a[], int i, int j)
    {
        int maxsum = -INT_MAX;
        int sum = 0;
        for (int k = i; k <= j; k++)
        {
            sum += a[k];
            maxsum = max(sum, maxsum);
        }
        return maxsum;
    }
    int maxsub(int a[], int i, int j)
    {
        if (i == j)
            return a[i];
        else
        {
            int m = (i + j) / 2;
            int wl = maxsub(a, i, m);
            int wr = maxsub(a, m + 1, j);
            int wm = maxsubleft(a, i, m) + maxsubright(a, m + 1, j);
            return max(wl, wr, wm);
        }
    }
    void timchiso(int a[], int i, int j, int &m, int &n)
    {
        int d = j - i;
        int sub = maxsub(a, i, j);
        for (int i = 0; i <= d; i++)
        {
            for (int j = i; j <= d; j++)
            {
                int s = 0;
                for (int k = i; k <= j; k++)
                {
                    s += a[k];
                    if (s == sub)
                    {
                        m = i;
                        n = j;
                        return;
                    }
                }
            }
        }
    }
}

```



```

    }
}
void main()
{
    int m = 0, n = 0;
    int a[] = { 13,-15,2,18,4,8,0,-5,-8 };
    timchiso(a, 0, 8, m, n);
    //printf("%d", m);
    for (int i = m; i <= n; i++)
    {
        printf("%5d", a[i]);
    }
    system("pause");
}

```

Bài toán hoán đổi phần tử của mảng

```

#include<iostream>
using namespace std;
void hoanvi(int &a, int &b)
{
    int c=a;
    a = b;
    b = c;
}
void Exchange(int a[], int i, int j, int leght)
{
    for (int k = 0;k < leght;k++)
    {
        hoanvi(a[i + k], a[j + k]);
    }
}
void Transpose(int a[],int l, int n, int m)
{
    int i = m, j = n-l+1-m; m = m + 1;
    while (i!=j)
    {
        if (i > j)
        {
            Exchange(a, m - i, m, j);
            i = i - j;
        }
        else
        {
            j = j - i;
            Exchange(a, m - i, m + j, i);
        }
    }
    Exchange(a, m - i, m, i);
}
void main()
{
    int a[] = { 1,2,3,4,5,6,7,8 };
    Transpose(a, 1,7,3);
    for (int i = 0;i < 8;i++)
    {

```

```

        cout << a[i]<<" ";
    }
    system("pause");
}

```

Bài toán tính lũy thừa

```

#include<iostream>
using namespace std;

long power(int a, int n)
{
    if (n == 0)
        return 1;
    else if (n % 2 == 1)
        return a*power(a*a, n / 2);
    else
    {
        return
            power(a*a, n / 2);
    }
}

void main()
{
    cout << power(9, 14);
    system("pause");
}

```

Bài toán dijkstra

```

#include<iostream>
#include<fstream>
using namespace std;
fstream f;
int a[100][100];
int Ddnn[100];
int i, k, Dht, MIN;
int VC = INT16_MAX;
int Daxet[100];
int L[100];
int n;
void Init()
{
    f.open("Input.txt");
    f >> n;
    for (int i = 1; i <= n; i++)
        for (int j = 1; j <= n; j++)
        {
            f >> a[i][j];
        }
    f.close();
}

void Xuatdd(int s, int k, int Ddnn[100])

```

```

{
    int i;
    cout << "\nDuong di ngan nhat tu" << s << "den" << k << "la :";
    i = k;
    while (i != s)
    {
        cout << i << "<--";
        i = Ddnn[i];
    }
    cout << s<<endl;
}

void Dijkstra(int s)
{
    for (i = 1; i <= n; i++)
    {
        Daxet[i] = 0;
        L[i] = VC;
    }
    Daxet[s] = 1;
    L[s] = 0;
    Dht = s;
    int j = 1;
    while (j <= n - 1)
    {
        MIN = VC;
        for (i = 1; i <= n; i++)
        {
            if (!Daxet[i])
            {
                if (L[Dht] + a[Dht][i] < L[i])
                {
                    L[i] = L[Dht] + a[Dht][i];
                    Ddnn[i] = Dht;
                }
                if (L[i] < MIN)
                {
                    MIN = L[i];
                    k = i;
                }
            }
        }
        Xuatdd(s, k, Ddnn);
        cout << "\nTrong so: " << L[k];
        Dht = k;
        Daxet[Dht] = 1;
        j++;
    }
}

void main()
{
    Init();
    Dijkstra(2);
    system("pause");
}

```

Bài toán prim

```
#include<iostream>
#include<fstream>
using namespace std;
fstream f;
int n;//số đỉnh
int a[100][100];//ma trận trọng số
int Closest[100];//Mảng lưu giá trị của đỉnh gần kề đỉnh đang xét;
int Lowcost[100];//Mảng lưu trọng số của cạnh(i,Closest[i])
void Init()
{
    f.open("Prim.txt");
    f >> n;
    for (int i = 1;i <= n;i++)
        for (int j = 1;j <= n;j++)
        {
            f >> a[i][j];
        }
    f.close();
}
void Prim(int a[][100])
{
    int i, j, k, Min;
    for ( i = 2;i <= n;i++)
    {
        Lowcost[i] = a[1][i];
        Closest[i] = 1;
    }
    for (i = 2;i <= n;i++)
    {
        Min = Lowcost[2];
        k = 2;
        for (j = 3;j <= n;j++)
        {
            if (Lowcost[j] < Min)
            {
                Min = Lowcost[j];
                k = j;
            }
        }
        cout << endl << k << "- "<< Closest[k];
        Lowcost[k] = INT16_MAX;
        for (j = 2;j <= n;j++)
        {
            if (a[k][j] < Lowcost[j] && Lowcost[j] < INT16_MAX)
            {
                Lowcost[j] = a[k][j];
                Closest[j] = k;
            }
        }
    }
}
void main()
{
    Init();
```

```

        Prim(a);
        system("pause");
    }

```

Bài toán tô màu đồ thị

```

#include<iostream>
using namespace std;

int a[100][100]; //Luu mang dau vao
int n; // So dinh
int sm=0; //so mau su dung, ban dau gan =0;roi tang dan len;
int x[100]; //luu ket qua//

void Nhap()
{
    cout << "Nhap vao n :";
    cin >> n;
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= n; j++)
        {
            cout << "a[" << i << "][" << j << "] = ";
            cin >> a[i][j];
        }
    }
}

void ToMau()
{
    for (int i = 1; i <= n; i++)
    {
        if (x[i] == 0)
        {
            sm++;
            x[i] = sm;
            for (int j = i + 1; j <= n; j++)
            {
                if (a[i][j] == 0 && x[j] == 0) //tim cac dinh khong ke vs dinh
                dang xet, va xem dinh do da to mau hay chua
                {
                    int kt = 1;
                    for (int k = i + 1; k < j; k++)
                    {
                        if (a[k][j] == 1 && x[i] == x[k]) //Kiem tra xem
                        mau cua dinh duoc to co trung mau vs cac dinh ke vs no k, neu trung thi bo qua
                        {
                            kt = 0;
                            break;
                        }
                    }
                    if (kt)
                        x[j] = sm;
                }
            }
        }
    }
}

```

```

    }
}
void xuatKq() { //In ket qua ra man hinh
    for (int i = 1; i <= sm; i++) {
        cout << "Mau " << i << ": ";
        for (int j = 1; j <= n; j++) if (x[j] == i) cout << j << " ";
        cout << endl;
    }
}
void main()
{
    Nhap();
    ToMau();
    xuatKq();
    system("pause");
}

```

Bài toán tìm dãy con có tổng lớn nhất

```

#include<conio.h>
#include<stdio.h>

int main() {

    int a[10] = { 0,13,-15,2,18,4,8,0,-5,-8 };

    int maxS = a[1], maxE = a[1], s1 = 1, s = 1, e = 1;
    for (int i = 2; i <= 9; i++) {
        if (maxE > 0) {
            maxE = maxE + a[i];
        }
        else {
            maxE = a[i];
            s1 = i;
        }
        if (maxE > maxS) {
            maxS = maxE;
            e = i;
            s = s1;
        }
    }
    printf("\n");
    printf("Day con lien tiep co Tong lon nhat: \n");
    printf("\n\n%d %d %d\n\n", maxS, s, e);

}

```

Bài toán Floy

```

#include<iostream>
#include<fstream>
using namespace std;
fstream f;
int n;
int c[100][100];
int d[100][100];
int p[100][100];
int k,h,m;

void Nhap()
{
    f.open("Floy.txt");
    f >> n;
    for (int i = 1;i <= n;i++)
        for (int j = 1;j <= n;j++)
        {
            f >> c[i][j];
        }
    f.close();
}

void Floyd()
{
    int i, j;
    for(i=1;i<=n;i++)
        for (j = 1;j <= n;j++)
        {
            d[i][j] = c[i][j];
            p[i][j] = 0;
        }
    for (k = 1;k <= n;k++)
    {
        for (i = 1;i <= n;i++)
        {
            if (d[i][k] > 0)
            {
                for (j = 1;j <= n;j++)
                {
                    if (d[k][j] > 0)
                    {
                        if (d[i][k] + d[k][j] < d[i][j])
                        {
                            d[i][j] = d[i][k] + d[k][j];
                            p[i][j] = k;
                        }
                    }
                }
            }
        }
    }
}

void Xuat(int s, int k)
{
    cout << "Nhap vao dinh xuat phat: ";
    cin >> s;
    cout << "Nhap vao dinh den :";
    cin >> k;
}

```

```

int i = k;
if (p[s][k] != 0)
{
    cout << "Duong di ngan nhat tu " << s << " den " << k << " la: ";
    while (i != 0)
    {
        cout << i << "<--";
        i = p[s][i];
    }
    cout << s<<endl;
    cout << "Trong so : " << d[s][k]<<endl;
}
else
{
    cout << "Duong di ngan nhat tu " << s << " den " << k << " la: " << k <<
"<--" << s;
    cout << "Trong so : " << d[s][k] << endl;
}

}
void main()
{
    Nhap();
    Floyd();
    Xuat(h,m);
    system("pause");
}

```

Bài toán Xâu con chung dài nhất

```

#include<iostream>
#include<stdio.h>
using namespace std;

int max(int a, int b)
{
    return a > b ? a : b;
}

void main()
{
    int F[100][100];
    int n, m;
    char x[100];
    char y[100];
    cin >> x;
    cin >> y;
    n = strlen(x);
    m = strlen(y);
    for (int i = 0; i <= n - 1; i++)
    {
        for (int j = 0; j < m - 1; j++)
        {

```



```

        F[i][j] = 0;
    }
}
for (int i = 1; i <= n-1; i++)
    for (int j = 1; j <= m-1; j++)
        if (x[i] == y[j])
            F[i][j] = F[i - 1][j - 1] + 1;
        else
            F[i][j] = max(F[i - 1][j], F[i][j - 1]);
printf("%d", F[m-1][n-1]);
system("pause");
}

```

Bài toán cái túi theo phương pháp quay lui

```

#include<iostream>
using namespace std;
int v[100]; //Mảng lưu giá trị của các túi
int w[100]; //Mảng lưu trọng lượng của các túi
int n; // Số đồ vật
int m; //Trọng lượng mà túi chứa được
int x[100]; //Lưu kết quả;
int Tl = 0; //Lưu trọng lượng mang được
int S = 0; //Giá trị của túi thu được sau từng bước;
int f = 0;
int x1[100];

void Nhap()
{
    cout << "Nhập vào số đồ vật : "; cin >> n;
    cout << "Nhập vào kích thước túi : "; cin >> m;
    cout << "Nhập vào giá trị của từng túi: " << endl;
    for (int i = 1; i <= n; i++)
    {
        cout << "v[" << i << "] = ";
        cin >> v[i];
    }
    cout << "Nhập vào trọng lượng của từng túi: " << endl;
    for (int i = 1; i <= n; i++)
    {
        cout << "w[" << i << "] = ";
        cin >> w[i];
    }
}

void Try(int i)
{
    int j, t, g;
    t = (int)(m - Tl) / w[i]; //tìm ra số đồ vật thu i mang đi tối đa được bao nhiêu
    for (j = t; j >= 0; j--)
    {
        x[i] = j;
        Tl = Tl + w[i] * x[i];
        S = S + v[i] * x[i];
        if (i == n)

```

```

        {
            if (S > f)
            {
                for (int k = 1; k <= n; k++)
                {
                    x1[k] = x[k];
                }
                f = S;
            }
        }
    else
    {
        /*g = S + v[i + 1] * (m - T1)/w[i + 1];
        if(g>f)*/
        //bài toán nhanh cần thi thêm 2 dòng code 53,54
        Try(i + 1);
    }
    T1 = T1 - w[i] * x[i];
    S = S - v[i] * x[i];
}
}
void Xuat()
{
    cout << "Nghiem La: ";
    for (int i = 1; i <= n; i++)
    {
        cout << x1[i];
    }
}
void main()
{
    Nhap();
    Try(1);
    Xuat();
    system("pause");
}

```

Bài toán liệt kê hoán vị(slide 12)

```

#include<iostream>

using namespace std;
int a[100], chuaxet[100], n;
void Xuat(int a[])
{
    for (int i = 1; i <= n; i++)
    {
        cout << a[i];
    }
    cout << endl;
}

void Init()
{

```

```

        for (int i = 1; i <= n; i++)
            chuaxet[i] = 1;
    }

    void Try(int i)
    {
        for (int j = 1; j <= n; j++)
        {
            if (chuaxet[j])
            {
                a[i] = j;
                chuaxet[j] = 0;
                if (i == n)
                {
                    Xuat(a);
                }
                else
                {
                    Try(i + 1);
                }
                chuaxet[j] = 1;
            }
        }
    }
}

int main()
{
    n = 6;
    Init();
    Try(1);
    system("pause");
}

```