Ôn tập UDTT

Contents

[Dạng 1: Các bài toán sinh 2](#_Toc120983491)

[-Sinh dãy nhị phân: vd n=3 thì sinh ra dãy 000, 001, 010,… 2](#_Toc120983492)

[-Liệt kê tập con: vd n=5, k=3 thì sinh ra 123,124,… 3](#_Toc120983493)

[-Liệt kê hoán vị: vd n = 6 sỉnh ra dãy 123456, 123465, … 5](#_Toc120983494)

[Dạng 2:kỹ thuật đệ quy khử đệ quy 7](#_Toc120983495)

[-Fibo, khử Fibo: 7](#_Toc120983496)

[-tháp hà nội 8](#_Toc120983497)

[-khử tháp HN 9](#_Toc120983498)

[-loang(Tìm miền liên thông) 12](#_Toc120983499)

[-Tìm UCLN 13](#_Toc120983500)

[-Tổ hợp chập k của n 14](#_Toc120983501)

[Dạng 3:chiến lược quay lui 14](#_Toc120983502)

[-Liệt kê hoán vị: 14](#_Toc120983503)

[-Mã đi tuần: 16](#_Toc120983504)

[-Tám hậu 18](#_Toc120983505)

[Dạng 4: Chiến lược tham lam 19](#_Toc120983506)

[-coin changing: 19](#_Toc120983507)

[-Bình nước 20](#_Toc120983508)

[-SX lịch 22](#_Toc120983509)

[-Người du lịch: 23](#_Toc120983510)

[Dạng 5: Chiến lược chia để trị: 23](#_Toc120983511)

[-TK nhị phân: 23](#_Toc120983512)

[-Tìm max: 24](#_Toc120983513)

[-Tính a^n 24](#_Toc120983514)

[-SX phân đoạn: 25](#_Toc120983515)

[Dạng 6: chiến lược quy hoạch động: 26](#_Toc120983516)

[-phân tích số thành tổng 26](#_Toc120983517)

[-dãy con đơn điệu 27](#_Toc120983518)

[-bài toán cái túi 28](#_Toc120983519)

[-bài toán cắm hoa 29](#_Toc120983520)

[Dạng 7:Các thuật toán xử lý xâu: 32](#_Toc120983521)

[-liệt kê các xâu con có độ dài k từ sâu gốc T nào đó 32](#_Toc120983522)

[-Kiểm tra p có là dãy con trong t hay không? 34](#_Toc120983523)

[-Thuật toán lặp 34](#_Toc120983524)

[-Thuật toán horspool 35](#_Toc120983525)

[-kiểm tra xâu p xuất hiện bao nhiêu lần trong t 35](#_Toc120983526)

[-tìm xâu con chung dài nhất 36](#_Toc120983527)

[Tìm min chia để trị 38](#_Toc120983528)

[====chọn 4 sinh viên từ 6 sinh viên==== 38](#_Toc120983529)

[========Tìm kiếm vị trí ch có trong dãy========== 41](#_Toc120983530)

# Dạng 1: Các bài toán sinh

## -Sinh dãy nhị phân: vd n=3 thì sinh ra dãy 000, 001, 010,…

CODE:

#include <iostream>

using namespace std;

int a[5];

void view\_config(int n){

for(int i=1; i <= n; i++)

cout << a[i];

cout<<endl;

}

void next\_config(int n, int i){

a[i]=1;

i++;

while(i<=n){

a[i]=0;

i++;

}

}

void config(int n){

a[n] = {0};

int i=1;

do{

view\_config(n);

i=n;

while(i>0 && a[i] == 1){

i--;

}

if(i>0) next\_config(n, i);

}while(i>0);

}

int main()

{

int n=5;

config(n);

return 0;

}

## -Liệt kê tập con: vd n=5, k=3 thì sinh ra 123,124,…

CODE:

#include <iostream>

using namespace std;// vd n = 5 k=3 thi sinh ra day 123 124 125 134 135.... day truoc vừa đủ lớn hơn dãy sau

int a[5];

void view\_config(int k){

for(int i=1; i <= k; i++)

cout << a[i];

cout<<endl;

}

void next\_config(int k, int i){

a[i]+=1;

i++;

while(i<=k){

a[i]=a[i-1]+1;

i++;

}

}

void config(int n, int k){

int i;

for(int i=1; i<=k; i++)

a[i]=i;

do{

view\_config(k);

i=k;

while(i>0 && a[i] == n-k+i){

i--;

}

if(i>0) next\_config(k, i);

}while(i>0);

}

int main()

{

int n=5;

config(n, 3);

return 0;

}

## -Liệt kê hoán vị: vd n = 6 sỉnh ra dãy 123456, 123465, …

CODE

#include <iostream>

using namespace std;

int a[100];

void swap(int &a, int &b){

int tg=a;

a=b;

b=tg;

}

void view\_config(int k){

for(int i=1; i <= k; i++)

cout << a[i];

cout<<endl;

}

void next\_config(int n, int i){

int k=n;

while(a[k]<a[i]) k--;

swap(a[i], a[k]);

int j=n;

i++;

while(i<j){

swap(a[i], a[j]);

i++;

j--;

}

}

void config(int n){

int i;

for(int i=1; i<=n; i++)

a[i]=i;

do{

view\_config(n);

i=n-1;

while(i>0 && a[i] > a[i+1]){

i--;

}

if(i>0) next\_config(n, i);

}while(i>0);

}

int main()

{

int n=5;

config(6);

return 0;

}

# Dạng 2:kỹ thuật đệ quy khử đệ quy

## -Fibo, khử Fibo:

công thức: a[i] = a[i-1] + a[i-2];

vd: dãy 1 1 2 3 5 8,…

CODE:

#include <iostream>

using namespace std;

int Fibo(int n){

if(n < 3) return 1;

else return Fibo(n-2) + Fibo(n-1);

}

int khu\_Fibo(int n){

int f, f1 = 1, f2 = 1;

for(int i=3; i <= n; i++){

f = f1+f2;

f2 = f1;

f1 = f;

}

return f;

}

int main()

{

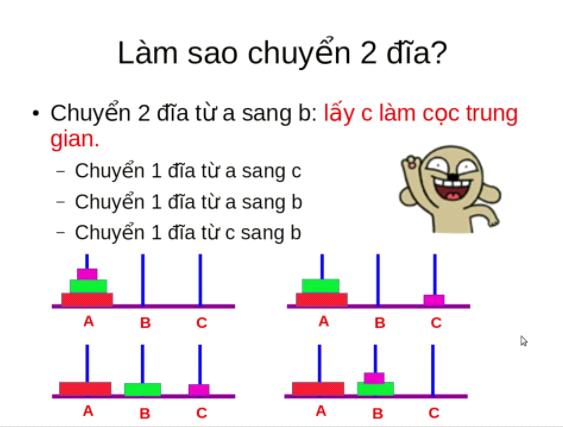
cout << Fibo(7) << endl;

cout << khu\_Fibo(7);

return 0;

}

## -tháp hà nội



CODE:

#include <iostream>

using namespace std;

void chuyen(int n, char A, char B, char C){

if(n == 1)

cout << "chuyen 1 dia tu coc " << A << " sang coc " << C << endl;

else{

chuyen(n-1, A, C, B);

chuyen(1, A, B, C);

chuyen(n-1, B, A, C);

}

}

int main()

{

chuyen(3, 'A', 'B', 'C');

return 0;

}

## -khử tháp HN

CODE:

#include <bits/stdc++.h>

using namespace std;

typedef struct{

int n;

char a, b, c;

}Call;

typedef struct{

int top;

Call \*a;

}Stack;

void create(Stack &s){

s.top = -1;

s.a = new Call[1];

}

bool empty(Stack s){

return (s.top == -1);

}

void push(Stack &s, Call call){

s.top++;

s.a = (Call\*) realloc(s.a, (s.top+1)\*sizeof(Call));

s.a[s.top] = call;

}

void pop(Stack &s, Call &call){

call = s.a[s.top];

s.a = (Call\*) realloc(s.a, (s.top)\*sizeof(Call));

s.top--;

}

int main()

{

int n=5, k;

Stack s;

char a, b, c;

Call call = {n, 'A', 'B', 'C'};

create(s);

push(s, call);

do{

pop(s, call);

a = call.a; b = call.b; c = call.c;

if(call.n == 1)

cout << "chuyen 1 dia tu coc " << a << " sang coc " << c << endl;

else{

k=call.n-1;

call.n = k;

call.a = b;

call.b = a;

call.c = c;

push(s, call);

call.n = 1;

call.a = a;

call.b = b;

call.c = c;

push(s, call);

call.n = k;

call.a = a;

call.b = c;

call.c = b;

push(s, call);

}

}while(!empty(s));

return 0;

}

## -loang(Tìm miền liên thông)

Vd ta có mảng : 1 1 1 1 1 ta có 3 miền liên thông với nhau là 1 1 1 1 1, 0 0 0 00

0 0 0 0 0

1 1 1 1 1

CODE:

#include <iostream>

using namespace std;

int arr[5][6] = {{1, 0, 1, 0, 1, 0},

{0, 0, 1, 0, 1, 0},

{1, 0, 1, 0, 1, 0},

{1, 0, 1, 0, 1, 0},

{1, 0, 1, 0, 1, 0}};

bool flag[5][6];

int so\_mien;

void loang(int i, int j){

flag[i][j] = true;

if(i > 0 && (arr[i-1][j] == arr[i][j]) && (!flag[i-1][j]))

loang(i-1, j);

if(i < 4 && (arr[i+1][j] == arr[i][j]) && (!flag[i+1][j]))

loang(i+1, j);

if(j > 0 && (arr[i][j-1] == arr[i][j]) && (!flag[i][j-1]))

loang(i, j-1);

if(j < 5 && (arr[i][j+1] == arr[i][j]) && (!flag[i][j+1]))

loang(i, j+1);

}

void demmien(){

for(int i=0; i < 5; i++)

for(int j=0; j < 6; j++){

if(!flag[i][j])

so\_mien++;

loang(i, j);

}

}

int main()

{

demmien();

cout << so\_mien;

return 0;

}

## -Tìm UCLN

CODE:

#include <iostream>

using namespace std;

int UCLN(int a, int b){

if(a%b == 0) return b;

else return UCLN(b, a%b);

}

int main()

{

cout << UCLN(4, 8) << endl;

return 0;

}

## -Tổ hợp chập k của n

CODE:

#include <iostream>

using namespace std;

int C(int k, int n){

if(n==k || k==0) return 1;

else return C(k-1, n-1) + C(k, n-1);

}

int main()

{

cout << C(2, 4);

return 0;

}

# Dạng 3:chiến lược quay lui

## -Liệt kê hoán vị:

CODE:

#include <iostream>

using namespace std;

int n, \*x, \*dd;

void xuat(){

for(int i=1; i <= n; i++)

cout << x[i];

cout <<endl;

}

void truyy(int k){

for(int i=1; i <= n; i++)

if(dd[i] == 0){

x[k] = i;

if(k == n)

xuat();

else{

dd[i] = 1;

truyy(k+1);

dd[i] = 0;

}

}

}

int main()

{

n=3;

dd = new int[n+1];

x = new int[n+1];

for(int i=0; i <= n; i++)

dd[i] = 0;

truyy(1);

return 0;

}

-Sinh dãy nhị phân:

(ae nên học theo phương pháp sinh cho dễ nhớ)

## -Mã đi tuần:

CODE:

#include <iostream>

using namespace std;

int a[9] = {0, 2, 1, -1, -2, -2, -1, 1, 2};

int b[9] = {0, 1, 2, 2, 1, -1, -2, -2, -1};

int \*\*H;

int n;

void Try(int k, int x, int y, int &tm){

int u, v, tm1, i=0;

do{

i++;

tm1 = 0;

u = x + a[i];

v = y + b[i];

if(1 <= u && u <= n && 1 <= v && v <= n && H[u][v] == 0){

H[u][v] = k;

if(k < n\*n){

Try(k+1, u, v, tm1);

if(!tm1)

H[u][v] = 0;

}

else tm1 = 1;

}

}while(!tm1 && (i < 8));

tm = tm1;

}

int main()

{

int x, y, suc;

n = 5;

H = new int\*[n+1];

for(int i=1; i <= n ;i++)

H[i] = new int[n+1];

for(int i=1; i <= n; i++)

for(int j=1; j <= n; j++)

H[i][j] = 0;

x = 1;

y = 1;

H[x][y] = 1;

Try(2, x, y, suc);

if(suc)

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

for(int j=1; j <= n; j++)

cout << H[i][j] << "\t";

cout << endl << endl;

}

else cout << "khong tim thay nuoc di";

return 0;

}

## -Tám hậu

CODE:

#include <iostream>

using namespace std;

int a[20];

bool Safety(int k, int i){

for(int j = 1; j < k; j++)

if(a[j] == i || abs(j-k) == abs(a[j] - i))

return false;

return true;

}

void view(int n){

for(int i=1; i <= n; i++)

cout << a[i];

cout << endl;

}

void Try(int k, int n){

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

if(Safety(k, i)){

a[k] = i;

if(k == n)

view(n);

else

Try(k+1, n);

}

}

}

int main()

{

int n = 8;

Try(1, n);

return 0;

}

# Dạng 4: Chiến lược tham lam

## -coin changing:

CODE:

#include <iostream>

using namespace std;

bool c\_A2(int c[], int m, long n, int s[]){

for(int i=0; i < m; i++)

s[i] = 0;

int i = 0;

while(n > 0 && i < m){

while(n - c[i] >= 0){

s[i]++;

n = n - c[i];

}

i++;

}

if(n > 0) return false;

else return true;

}

int main()

{

int c[10] = {100, 25, 10, 5, 1};

int s[10];

int n = 340, m = 5;

bool a = c\_A2(c, 5, n, s);

if(a){

cout << "result" << endl;

for(int i=0; i < m; i++){

cout << "\t" << c[i] << ":" << s[i] << endl;

}

}

else

cout << "false";

return 0;

}

## -Bình nước

CODE:

#include <iostream>

using namespace std;

int c[10] = {100, 25, 10, 5, 1};

int s[10];

int n = 6, m = 5;

int binhnuoc(){

for(int i=0; i < m; i++)

s[i] = 0;

int i = m-1, dem=0;

while(n > 0 && i >= 0){

if(n - c[i] >= 0){

s[i]++;

n=n-c[i];

dem++;

i--;

}

else break;

}

return dem;

}

int main()

{

cout << binhnuoc() << endl;

cout << "result" << endl;

for(int i=0; i < m; i++)

cout << "\t" << c[i] << ":" << s[i] << endl;

return 0;

}

## -SX lịch

CODE:

#include <iostream>

using namespace std;

void sx(){

// sap xem giam dan theo 1 gia tri vd a = 1 2 3 2 4 1 2 3 4 2

// b = 1 3 2 1 2 sap mang a theo b 1 1 2 2 3

}

float\* sxLich(int n, float s[], float F[], float a[]){

a[n] = {0};

float L\_F = 0;

for(int i=0; i < n; i++){

if(s[i] >= L\_F){

a[i]++;

L\_F = F[i];

}

}

return a;

}

int main()

{

float S[5] = {8, 9, 10, 11, 12};

float F[5] = {8.5, 11, 11.5, 12.5, 13};

int n = 5;

float \*a = new float[5];

a = sxLich(n, S, F, a);

for(int i=0; i <= n; i++)

if(a[i] == 1)

cout << i + 1 << " ";

return 0;

}

## -Người du lịch:

(khó vl nên ae thông cảm nhé)

# Dạng 5: Chiến lược chia để trị:

## -TK nhị phân:

CODE:

#include <iostream>

using namespace std;

int a[7] = {1, 2, 3, 4, 5, 6, 7};

int tknp(int l, int r, int x){

if(l<=r){

int mid = (l + r)/2;

if(a[mid] == x) return mid;

if(a[mid] < x) return tknp(mid+1, r, x);

else return tknp(l, mid-1, x);

}

return -1;

}

int main()

{

cout << tknp(0, 6, 1);

return 0;

}

## -Tìm max:

#include <iostream>

using namespace std;

int a[7] = {1, 2, 3, 10, 5, 6, 7};

int Max(int l, int r){

int m = 0;

if(l==r) m = a[l];

else{

int tg = (l+r)/2;

int temp1 = Max(l, tg);

int temp2 = Max(tg+1, r);

m=temp1>temp2?temp1:temp2;

}

return m;

}

int main()

{

cout << Max(0, 6);

return 0;

}

## -Tính a^n

CODE:

#include <iostream>

using namespace std;

int pow(int a,int n){

if(n==1) return a;

int tmp=pow(a, n/2);

return (n&1)?tmp\*tmp\*a:tmp\*tmp;

}

int main()

{

cout << pow(3, 3);

return 0;

}

## -SX phân đoạn:

CODE:

#include <iostream>

using namespace std;

int a[5] = {1, 3, 2, 4, 2};

int sxphandoan(int l, int r){

if(l<r){

int i=l;

int j=r;

int mid=(l+r)/2;

int chot = a[mid];

do{

while(a[i] < chot) i++;

while(a[j] > chot) j--;

if(i<=j){

int tg=a[i];

a[i] = a[j];

a[j] = tg;

i++;

j--;

}

}while(i<=j);

sxphandoan(l, j);

sxphandoan(i, r);

}

}

int main()

{

sxphandoan(0, 4);

for(int i=0;i<5;i++)

cout <<a[i];

return 0;

}

# Dạng 6: chiến lược quy hoạch động:

## -phân tích số thành tổng

CODE:

#include <iostream>

using namespace std;

int F[10], n=6;

int analys03(){

F[0] = 1;

for(int i=1; i <=n; i++)

F[i] = 0;

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

for(int j=i; j<=n; j++){

F[j] = F[j] + F[j-i];

}

}

return F[n];

}

int main()

{

cout << analys03() << endl;

return 0;

}

## -dãy con đơn điệu

CODE:

#include <iostream>

using namespace std;

int a[9] = {1, 2, 3, 4, 9, 10, 5, 6, 7}, L[9], n = 9, T[9];

void dp\_alg(){

a[0] = -12345674;

a[n+1] = 12345674;

L[n+1] = 1;

for(int i=n; i >= 0; i--){

int jmax = n + 1;

for(int j=i+1; j <= n+1; j++)

if(a[j] > a[i] && L[j] > L[jmax])

jmax = j;

L[i] = L[jmax] + 1;

T[i] = jmax;

}

}

void dp\_res(){

int k = 0;

cout << "Day con:{";

while(k!=n+1){

cout << a[k] << " ";

k = T[k];

}

cout << "}";

}

int main()

{

dp\_alg();

dp\_res();

return 0;

}

## -bài toán cái túi

CODE:

#include <iostream>

// xem so gia tri co the bo vao cai tui là max khong vuot qua n gia tri

using namespace std;

int C\_a(int c[], long n, int m){

int i = m - 1;

int Count = 0;

while(n > 0 && i >= 0){

if(n > c[i]){

Count++;

}

n = n - c[i];

i--;

}

return Count;

}

int main()

{

int c[10] = {8, 5, 4, 3, 2}, n = 20;

int dem = C\_a(c, n, 5);

cout << dem;

return 0;

}

## -bài toán cắm hoa

CODE:

#include <iostream>

using namespace std;

int n=10, k=6;

int X[7][11] = {{0,0,0,0,0,0,0,0,0,0,0},

{0,1, 20, 2, 3, 4, 5, 6, 7, 8, 9},

{0,2, 50, 3, 4, 5, 6, 9, 8, 7, 10},

{0,5, 8, 10, 12, 1, 2, 3, 4, 5, 6},

{0,3, 4, 5, 4, 3, 2, 1, 5, 3, 2},

{0,6, 8, 5, 9, 5, 2, 1, 4, 5,6},

{0,5, 3, 6, 2, 4, 8, 9, 5, 2, 1}};

int L[7][11]={0};

void PA()

{

for(int i=1; i<=k; i++)

{

for(int j=1; j<=n; j++)

{

if(i==j) L[i][j] = L[i-1][j-1] + X[i][j];

else if(i<j)

{

L[i][j] = L[i][j-1];

if(L[i][j] < L[i-1][j-1]+X[i][j])

L[i][j] = L[i-1][j-1] + X[i][j];

}

}

}

cout<<"\n\nPhuong an:\n\n";

for(int i=1; i<=k; i++)

{

for(int j=1; j<=n; j++)

{

cout<<L[i][j]<<" ";

}

cout<<endl;

}

cout<<"\n\ngia tri tham mi lon nhat: "<<L[k][n];

int i=k;

cout<<"\ncac bong hoa duoc cam vao lo theo so hieu tu lon den nho: ";

for(int j=n; j>=1; j--)

{

if(L[i][j] != L[i][j-1])

{

cout<<j<<" ";

i--;

}

}

}

int main()

{

for(int i=1; i<=k; i++)

{

for(int j=1; j<=n; j++)

{

cout<<X[i][j]<<" ";

}

cout<<endl;

}

cout <<endl;

for(int i=1; i<=k; i++)

{

for(int j=1; j<=n; j++)

{

cout<<L[i][j]<<" ";

}

cout<<endl;

}

PA();

return 0;

}

# Dạng 7:Các thuật toán xử lý xâu:

## -liệt kê các xâu con có độ dài k từ sâu gốc T nào đó

CODE:

#include <iostream>

#include <string.h>

using namespace std;

char \*s = "abccaba";

int k=3, dem=0, n;

char xc[10];

bool comat(char ch, char \*s){

int i=0;

while(s[i] != '\0'){

if(s[i] == ch) return true;

i++;

}

return false;

}

void them(char \*s, char ch){

int n=strlen(s);

s[n]=ch;

s[n+1] = '\0';

}

void xoa(char \*s){

int n=strlen(s);

s[n-1]='\0';

}

void th(int i){

char tam[1000];tam[0]='\0';

if(strlen(xc)==k){

cout <<xc<<endl;

dem++;

}

else{

for(int j=i;j<n;j++){

if(comat(s[j], tam) == false){

them(tam, s[j]);

them(xc, s[j]);

th(j+1);

xoa(xc);

}

}

}

}

int main()

{

n=6;

th(0);

return 0;

}

## -Kiểm tra p có là dãy con trong t hay không?

### -Thuật toán lặp

CODE:

#include <iostream>

#include <string.h>

using namespace std;

int indexOf(const char \*p, const char \*t){

int m = strlen(p);

int n = strlen(t) - m;

for (int i = 0; i < n; i++){

int j = 0;

while (j < m && t[i + j] == p[j]){

j++;

}

if (j == m){

return i;

}

}

return -1;

}

int main()

{

char \*p="cong hoa xa hoi chu nghia";

char \*t="hoa";

//neu tra ve -1 thi t khong xuat hien trong t

cout << indexOf(t, p);

return 0;

}

### -Thuật toán horspool

(ae dung phuong phap z cho de nho)

## -kiểm tra xâu p xuất hiện bao nhiêu lần trong t

(Thuat toan Z)

CODE:

#include <iostream>

#include <string.h>

using namespace std;

void view(int \*z, int n){

for(int i=0; i < n; i++)

cout << z[i] << " ";

}

void Z(const char \*s, int z[]){

int n = strlen(s),l = 0, r = 0;

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

if(i>r){

l=r=i;

while(r<n && s[r-i] == s[r]) r++;

z[i] = r-l; r--;

}else if(z[i-l] < r-i+1)

z[i] = z[i-l];

else{

l=i;

while(r<n && s[r-l] == s[r]) r++;

z[i] = r-l; r--;

}

}

view(z, n);

}

int main()

{

char \*s = "vietnam$xaviet viet hoivietchu nghia";

int n=strlen(s);

int z[n];

for(int i=0; i < n; i++)

z[i] = 0;

Z(s, z);

return 0;

}

## -tìm xâu con chung dài nhất

#include <iostream>

#include <string.h>

using namespace std;

int L[50][50];

void alog\_1(char \*a, char \*b){

for(int i=0; i < strlen(a); i++)

for(int j=0; j < strlen(b); j++)

L[i][j] = 0;

for(int i=0; i < strlen(a); i++)

for(int j=0; j < strlen(b); j++){

if(a[i] == b[j])

L[i][j] = L[i-1][j-1] + 1;

else

L[i][j] = L[i-1][j]>L[i][j-1]?L[i-1][j]:L[i][j-1];

}

for(int i=0; i < strlen(a); i++){

for(int j=0; j < strlen(b)-1; j++)

cout << L[i][j] << " ";

cout <<endl;

}

}

int main()

{

char \*a = "viettttt nammm";

char \*b = "viett ";

alog\_1(a, b);

return 0;

}

=============================================================

## Tìm min chia để trị

#include <iostream>

using namespace std;

int a[7] = {1, 2, 3, 10, 5, 6, 7};

int Min(int l, int r){

int m = 0;

if(l==r) m = a[l];

else{

int tg = (l+r)/2;

int temp1 = Min(l, tg);

int temp2 = Min(tg+1, r);

m=temp1<temp2?temp1:temp2;

}

return m;

}

int main()

{

cout << Min(0, 6);

return 0;

}

## ====chọn 4 sinh viên từ 6 sinh viên====

#include<bits/stdc++.h>

using namespace std;

int a[50];

void view\_config(int k)

{

for(int i=1; i<=k; i++)

{

if(a[i] == 1)

{

cout<<"Trang"<<"\t";

}

if(a[i] == 2)

{

cout<<"Tung"<<"\t";

}

if(a[i] == 3)

{

cout<<"Vu"<<"\t";

}

if(a[i] == 4)

{

cout<<"Kim"<<"\t";

}

if(a[i] == 5)

{

cout<<"Anh"<<"\t";

}

if(a[i] == 6)

{

cout<<"Cuong"<<"\t";

}

}

cout<<endl;

}

void next\_config(int i, int k)

{

a[i]+=1;

i++;

while(i<=k)

{

a[i] = a[i-1]+1;

i++;

}

}

void config(int n, int k)

{

int i;

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

{

a[i] =i;

}

do

{

view\_config(k);

i =k;

while(i>0 && a[i] == n-k+i)

{

i--;

}

if(i>0)

{

next\_config(i,k);

}

}

while(i>0);

}

int main()

{

int n, k;

cout<<"Nhap vao gia tri n = "; cin>>n;

cout<<"Nhap vao gia tri k = "; cin>>k;

config(n,k);

return 0;

}

## ========Tìm kiếm vị trí ch có trong dãy==========

#include<bits/stdc++.h>

using namespace std;

char a[7] = "abcdef";

char \*ch = "c";

int n = strlen(a);

int TKNP(int left, int right, char flag)

{

if(left <= right)

{

int mid = (left + right)/2;

if(a[mid] == flag)

{

return mid;

}

if(a[mid] < flag)

{

return TKNP(mid+1, right, flag);

}

else

{

return TKNP(left, mid -1, flag);

}

}

return -1;

}

int main()

{

cout<<"Vi tri a["<<TKNP(0,n,\*ch)<<"] = "<<ch;

return 0;

}

Cho một dãy số gồm n(n<=1000) số nguyên dương A1, A2 ,… An và số nguyên dương k(k<=50). Hãy tìm dãy con gồm nhiều phần tử nhất của dãy đã cho sao cho tổng các phần tử của dãy con này chia hết cho k

Dãy số Wavio : là dãy số nguyên thỏa mãn các tính chất : các phần tử đầu sắp xếp thành 1 dãy tăng dần đến 1 phần tử đỉnh sau đó giảm dần. Ví dụ dãy số 1 2 3 4 5 2 1 là 1 dãy Wavio độ dài 7 . Cho 1 dãy gồm N số nguyên, hãy chỉ ra một dãy con Wavio có độ dài lớn nhất trích ra từ dãy đó

Dãy tam giác bao nhau : Cho n tam giác trên mặt phẳng, tam giác i bao tam giác j nếu 3 đỉnh của tam giác j đều nằm trong tam giác i( có thể nằm trên cạnh). Hãy tìm dãy tam giác bao nhau có nhiều tam giác nhất

Sinh chuỗi kí tự a,b

#include<iostream>

using namespace std;

const int N = 30;

int n;

char a[N];

void dq(int k)

{

if (k > n) {

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

cout << a[i];

}

cout << "\n";

return;

}

a[k] = 'a';

dq(k + 1);

a[k] = 'b';

dq(k + 1);

}

int main()

{

cin >> n;

dq(1);

return 0;

}

Bài 8 quân hậu

#include<algorithm>

#include<iostream>

#include<vector>

#include<string>

#include<stack>

#include<deque>

#include<set>

#include<map>

using namespace std;

const int N = 10;

int a[N],ans;

int cot[N], cheo1[2\*N], cheo2[2\*N];

void dequy(int k)

{

if (k > 8)

{

for (int i = 1;i <= 8;++i)

{

for (int j = 1;j <= 8;++j)

{

if (j == a[i])

cout << "|\*";

else

cout << "|.";

}

cout << "|"<<endl;

}

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;;

ans++;

return;

}

for (int i = 1;i <= 8;++i)

{

a[k] = i;

if (cot[i] || cheo1[k + i] || cheo2[k - i + 8])continue;

cot[i] = true;

cheo1[k + i] = true;

cheo2[k - i + 8] = true;

dequy(k + 1);

cot[i] = false;

cheo1[k + i] = false;

cheo2[k - i + 8] = false;

}

}

int32\_t main()

{

ios\_base::sync\_with\_stdio(0);

cin.tie(0);

cout.tie(0);

dequy(1);

cout << ans << endl;

return 0;

}

Bài 8 quân hậu đệ quy

#include<algorithm>

#include<iostream>

#include<vector>

#include<string>

#include<stack>

#include<deque>

#include<set>

#include<map>

using namespace std;

const int N = 10;

int a[N], ans;

bool cot[N], cheo1[2 \* N], cheo2[2 \* N];

bool kiemtra()

{

for (int i = 1;i <= 8;++i)

{

cot[i] = false;

cheo1[i] = false;

cheo2[i] = false;

cheo1[i + 8] = false;

cheo2[i + 8] = false;

}

for (int k = 1;k <= 8;++k)

{

int i = a[k];

if (cot[i] == true)return false;

if (cheo1[k+i] == true)return false;

if (cheo2[k-i+8] == true)return false;

cot[i] = true;

cheo1[k + i] = true;

cheo2[k - i + 8] = true;

}

return true;

}

void inketqua()

{

for (int i = 1;i <= 8;++i)

{

for (int j = 1;j <= 8;++j)

{

if (j == a[i])

cout << "|\*";

else

cout << "|.";

}

cout << "|" << endl;

}

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;;

ans++;

}

void dequy(int k)

{

if (k > 8 )

{

if (kiemtra() == true)

{

inketqua();

}

return;

}

for (int i = 1;i <= 8;++i)

{

a[k] = i;

dequy(k + 1);

}

}

int32\_t main()

{

ios\_base::sync\_with\_stdio(0);

cin.tie(0);

cout.tie(0);

dequy(1);

cout << ans << endl;

return 0;

}

Độ chênh lệch tổng 2 dãy

#include<algorithm>

#include<iostream>

#include<vector>

#include<string>

#include<stack>

#include<deque>

#include<set>

#include<map>

using namespace std;

const int N = 50;

int a[N];

int n,Min,cnt=0;

//Min là chênh lệch giữa 2 dãy

void nhap()

{

cin >> n;

for (int i = 1;i <= n;++i)

{

cin >> a[i];

Min += a[i];//ban đầu coi tất cả các phẩn tử đều thuộc cùng 1 dãy

}

}

// A là tổng dãy thứ nhất, B là tổng dãy thứ 2

void timMin(int i,int A,int B )

{

if (i > n){

Min = min(Min, abs(A-B) );

return;

}

timMin(i + 1, A+a[i],B);

timMin(i + 1, A,B + a[i]);

}

void truyvet(int i,int A,int B, string trangthai)

{

if (i > n)

{

if (Min == abs(A-B)) {

cnt++;

cout << "Cach thu : "<<cnt<<"\n";

cout << "Chi so cua cac so thuoc day 1 : ";

for (int k = 0;k < trangthai.size();++k)

{

if (trangthai[k] == '0')

cout << k+1 << " ";

//nêu in ra số thia a[k+1]

}

cout << "\n";

cout << "Chi so cua cac so thuoc day 2 : ";

for (int k = 0;k < trangthai.size();++k)

{

if (trangthai[k] == '1')

cout << k+1 << " ";

}

cout << "\n\n";

}

return;

}

truyvet(i+1,A+a[i],B, trangthai+"0");

truyvet(i+1,A,B+a[i], trangthai+"1");

}

int main()

{

nhap();

timMin(1,0,0);

cout << "Chenh lech nho nhat la: " << Min << "\n";

// để tránh 2 dãy hoán vị cho nhau coi dãy thứ nhất là dãy chứa phần tử a[1]

truyvet(2,a[1],0,"0");

/\*

\* nếu tinh cả 2 dãy hoán vị cho nhau

\* truyvet(1, 0, 0, "");

\*/

return 0;

}

Bài toán cái túi

#include <stdio.h>

#include <math.h>

#define MAX 100

int weight[MAX], value[MAX], n = 4, WW = 7, K[MAX][MAX];

int cmp(int a, int b)

{

return a > b ? a : b;

}

int solution()

{

K[n + 1][WW + 1];

for (int i = 0; i <= n; i++)

{

for (int w = 0; w <= WW; w++)

{

if (i == 0 || w == 0)

{

K[i][w] = 0;

}

else if (w < weight[i])

{

K[i][w] = K[i - 1][w];

}

else

{

K[i][w] = cmp(K[i - 1][w], value[i] + K[i - 1][w - weight[i]]);

}

}

}

return K[n][WW];

}

void printSolution()

{

// int i = n, j = WW;

// while (i != 0)

// {

// if (K[i][j] != K[i - 1][j])

// {

// printf("%3d", i);

// j = j - weight[i]; //lay goi thu i

// }

// i--;

// }

for (int i = 1; i <= n; i++)

{

for (int j = 1; j <= WW; j++)

{

printf("%3d", K[i][j]);

}

printf("\n");

}

}

int main()

{

for (int i = 1; i <= n; i++)

{

scanf("%d%d", &weight[i], &value[i]);

}

int ans = solution();

printf("Ans = %d\n", ans);

printSolution();

printf("\n");

}

// weight: 1 3 4 5

// value: 1 4 5 7