蓝桥杯知识点整理

2018年蓝桥杯b组题目与题解

https://blog.csdn.net/charles_zaqdt/article/details/79786821

复习大纲

1、C++ STL 常见算法

2、C++ 输入输出（包括流、文件）

3、C++常用泛型：list vector stack map

4、暴力穷举

5、递归

6、全排列 next_permutation 康托展开式

7、回溯

8、DFS、BFS、hash表

9.数学上的有：辗转相除（两行内），素数等

10.位运算

1.substr操作

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    ios::sync_with_stdio(false);
    string s="abcdefg";

    //s.substr(pos1,n)返回字符串位置为pos1后面的n个字符组成的串
    string s2=s.substr(1,5);//bcdef

    //s.substr(pos)//得到一个pos到结尾的串
    string s3=s.substr(4);//efg

    return 0;
}
```

2.全排列函数

头文件`#include<algorithm>`

bool next_permutation(iterator start, iterator end);

next_permutation函数的返回值是布尔类型

例：

```
#include<iostream>
#include<algorithm>
using namespace std;
int main(){
    string str="abc";
    while(next_permutation(str.begin(),str.end()))
        cout<<str<<endl;
    return 0;
}
```

```
#include<iostream>
#include<algorithm>
#include<vector>
using namespace std;
int main(){
    vector<int> dp;
    dp.push_back(1);
    dp.push_back(2);
    dp.push_back(3);
    while(next_permutation(dp.begin(),dp.end())){
        cout<<dp[0]<<dp[1]<<dp[2]<<endl;
    }
    return 0;
}
```

3.线性筛

```
int Mark[MAXSIZE];
int prime[MAXSIZE];

//判断是否是一个素数  Mark 标记数组 index 素数个数
int Prime(){
    int index = 0;
    memset(Mark,0,sizeof(Mark));
    for(int i = 2; i < MAXSIZE; i++)
    {
        //如果未标记则得到一个素数
        if(Mark[i] == 0){
            prime[index++] = i;
        }
        //标记目前得到的素数的i倍为非素数
        for(int j = 0; j < index && prime[j] * i < MAXSIZE; j++)
        {
            Mark[i * prime[j]] = 1;
```

```
            if(i % prime[j] == 0){
                break;
            }
        }
    }
    return index;
}
```

4.回文数判断

```
bool isPalindrome (int x) {
    int y = 0, z = x;
    while(x) {
        y = y * 10 + x % 10;
        x /= 10;
    }
    return z == y;
}
```

5.大数乘法

```


#include <cstdio>
#include <cmath>
#include <cstring>

void fan(char s[]) {
    char t;
    int i, j;
    for(i = 0,j = strlen(s) - 1; i <= j; i++, j--)
    {
        t = s[i];
        s[i] = s[j];
        s[j] = t;
    }
}

int main() {
    int p = 0, g = 0,h = 1;
    int k, l;
    char x[100010], y[100010], z[100010];
    while(scanf("%s %s",x,y) != EOF) {
        p = 0;
        fan(x);
        fan(y);
        k = strlen(x);
        l = strlen(y);
        int i;
        for(i = 0; i < k || i < l; i++) {
            if(i < k && i < l )
```

```
                z[i] = x[i] + y[i]+ p - '0';
            else if(i < k && i >= l)
                z[i] = x[i]+p;
            else if(i >= k && i < l)
                z[i] = y[i]+p;
            if(z[i] > '9')
            {
                z[i]-=10;
                p = 1;
            }
            else
                p = 0;
        }
        if(p)
            z[i++] = '1';
        z[i] = '\0';
        fan(x);
        fan(y);
        fan(z);
        printf("%s + %s = %s\n",x,y,z);
    }
    return 0;
}
```

6.二分法

```
int binary_search(int (*arr)(int), int x, int n) {
    int l = 0, r = n - 1, mid;
    while(l <= r) {
        mid = (l + r) >> 1;
        if(arr(mid) == x) return mid;
        if(arr(mid) > x) l = mid - 1;
        else r = mid + 1;
    }
    return -1;
}
```

7.最大公约数

```
#include<stdio.h>
void swap(int *m,int *n){
    int t;
    if(m < n){
        t = n;
        n = m;
        m = t;
    }
}

int  f(int m,int n){
```

```
    if(m % n == 0)return n;
    else return f(n,m % n);
}

int main(){
    int p = 0,m = 0,n = 0;
    printf("请按照由大到小的顺序输入两个整数,用空格隔开:\n");
    scanf("%d %d",&m,&n);
    swap(m,n);
    p = f(m,n);
    printf("两个数的最大公约数是:%d\n",p);
    return 0;
}
```

8.排序

插入

```
void insertion_sort (int a[], int n) {
    int i,j,v;
    for (i = 1; i < n; i++) {
         //如果第i个元素小于第j个，则第j个向后移动
        for (v = a[i], j = i - 1; j>=0 && v < a[j]; j--)
            a[j + 1] = a[j];
        a[j+1] = v;
    }
}
```

选择

```
void selection_sort (int a[], int n) {
    int i, j, pos, tmp;
    for (i = 0; i < n - 1; i++) {
         //寻找最小值的下标
        for (pos = i, j = i + 1; j < n; j++)
            if (a[pos] > a[j])
                pos = j;
        if (pos != i) {
            tmp = a[i];
            a[i] = a[pos];
            a[pos] = tmp;
        }
    }
}
```

冒泡

```
void bubble_sort (int a[], int n) {
```

```
    int i, j, lastSwap, tmp;
    for (j = n - 1; j > 0; j = lastSwap) {
        lastSwap=0;     //每一轮要初始化为0，防止某一轮未发生交换，lastSwap保留上一轮的值进入死循环
        for (i = 0; i < j; i++) {
            if (a[i] > a[i+1]) {
                tmp = a[i];
                a[i] = a[i+1];
                a[i+1] = tmp;
                 //最后一次交换位置的坐标
                lastSwap = i;
            }
        }
    }
}
```

快排

```
int mpartition(int a[], int l, int r) {
    int pivot = a[l];

    while (l < r) {
        while (l < r && pivot <= a[r]) r--;
        if (l<r) a[l++] = a[r];
        while (l < r && pivot > a[l]) l++;
        if (l < r) a[r--]=a[l];
    }
    a[l] = pivot;
    return l;
}

void quick_sort (int a[], int l, int r) {
    if (l < r) {
        int q = mpartition(a, l, r);
        msort(a, l, q - 1);
        msort(a, q + 1, r);
    }
}
```

9.汉诺塔(递归)

```
#include <iostream>

using namespace std;

void Hanno_Tower(int n, char a, char b, char c) { //这里代表将a柱子上的盘子借助b柱子移动到c柱子
    if(n == 1) {    //如果是一个盘子直接将a柱子上的盘子移动到c
        cout << a << "-->" << c << endl;
    } else {
        Hanno_Tower(n - 1, a, c, b);        //将a柱子上n-1个盘子借助c柱子，移动到b柱子
        cout << a << "-->" << c << endl;    //再直接将a柱子上的最后一个盘子移动到c
```

```
        Hanno_Tower(n - 1, b, a, c);        //然后将b柱子上的n-1个盘子借助a移动到c
    }
}

int main() {
    int n;
    cin >> n;
    Hanno_Tower(n, 'A', 'B', 'C');
    return 0;
}
```

10.斐波那契数

```
#include<stdio.h>
int main () {
    int f1 = 1, f2 = 1, n, i, f;
    scanf("%d", &n);
    if(n == 1 || n == 2) {
        f = 1;
    } else {
        f1 = 1;
        f2 = 1;
        for(i = 2 ; i < n ; i++) {
            f = (f1 + f2) % 1000000007;
            f1 = f2;
            f2 = f;
        }
    }
    printf("%d", f);
    return 0;
}
```

11.分解质因数

```
 #include<stdio.h>
 int main(){
     int a,b,i;
     int state;//状态
     int c;//记录当前的值
     scanf("%d %d",&a,&b);
     for(i = a;i <= b; i++){
         state = 1;
         for(int j = 2; j < i; j++)
             if(i % j == 0)  //若非质数让state为0
             {
                 state = 0;
                 break;
             }
         if(state == 1){//若为质数
             printf("%d = %d\n", i, i);
```

```
        }else if(state == 0){
            printf("%d = ", i);
            int j = 2;
            c = i;
            while(1) {
                while(c % j == 0) {
                    printf("%d", j);
                    c = c / j;
                    if(c != 1)
                        printf("*");
                }
                if(c == 1){
                    printf("\n");
                    break;
                }
                j++;
            }
        }
    }
}
```

12.十转任意进制

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
int TenNum(char a[],int B);              //将输入的数字转换成10进制数
void Numchange(int m, int B);            //将转换好了的10进制数转换为所需进制数
int TenNum(char a[], int B)
{
    int len, i, num;
    int sum = 0;
    len = strlen(a);                    //求得字符串长度
    for (i = 0; i < len; i++)
    {
        if (a[i] >= '0' && a[i] <= '9')
            num = a[i] - '0';
        else if (a[i] >= 'A' && a[i] <= 'F')
            num = a[i] - 'A' + 10;
        sum = sum * B + num;
    }
    return sum;
}
void Numchange(int m, int B)
{
    int n;
    if (m)
    {
        Numchange(m / B, B);
        n = m % B;
        if (n < 10)
```

```
            printf("%d", n);           //小于10直接输出
        else
            printf("%c", n + 55);      //大于10转换成字符输出
    }
}
int main()
{
    int B, b;
    char a[20];
    printf("请输入待转换数的进制（2-16）：");
    do {
        scanf_s("%d", &B);
    } while (B < 2 && B > 16);
    printf("请输入待转换数: ");
    getchar();
    gets_s(a);                          //将输入的n进制数存放在数组a中
    int m = TenNum(a, B);               //将输入的数字转换成十进制数
    printf("请输入需要转成几进制数（2-16）：");
    do {
        scanf_s("%d", &b);
    } while (B < 2 && B > 16);
    printf("%d进制数%s转换为%d进制数的结果为: ",B,a,b);
    Numchange(m, b);                   //将十进制数转换为所需进制数
    printf("\n");
    system("pause");
    return 0;
}
```