


原创 AICodeThunder 已于 2023-12-10 15:00:11 修改 阅读量126 收藏 点赞数 版权

分类专栏: C++知识点总结 文章标签: C/C++ 编程笔记 知识点总结

 C++知识点总结 专栏收录该内容 7 订阅 53 篇文章 订阅专栏

尺取法

- 一、复习枚举算法
1. 算法三要素

2. 最小公倍数公式

3. 时间复杂度
- 二、算法优化初级
1. 概念

2. 例题

(1) 最长小写字串

I 初步算法

II 认识尺取法

III 尺取法程序

(2) 最长递增子串

(3) 最小子串和

I 伪代码

II 完整代码

(4) 最短字符串包含

I 伪代码

II 代码

目录

尺取法

- 一、复习枚举算法
1. 算法三要素

2. 最小公倍数公式

3. 时间复杂度
- 二、算法优化初级
1. 概念

2. 例题

(1) 最长小写字串

(2) 最长递增子串

(3) 最小子串和

(4) 最短字符串包含

II 代码

一、复习枚举算法

1. 算法三要素

- 【枚举对象】要枚举的对象
- 【枚举范围】每一个枚举对象从几开始，到几结束
- 【筛选条件】筛选满足一定条件的数据

2. 最小公倍数公式

假如现在要求整数  $a$  和整数  $b$  的最小公倍数。  
求解公式如下：  
$$a \times b = gcd(a, b) \times lcm(a, b)$$

3. 时间复杂度

一般指程序运行的最大次数，不能超过  $10^8$ ，即  $1e8$ ，时间复杂度越高，程序运行时间越长。

二、算法优化初级

1. 概念

通过针对题型的方式，来使算法的某一方面变得更强的过程（包括用时更短、空间更小）。

2. 例题

(1) 最长小写字串

 AICodeThunder 关注

0

0

0

0

专栏目录

给定一个由若干大写小写字母组成的字符串 *str* , 现在请你求出 *str* 中最长的小写字串的长度。如果没有, 则输出 0。

【输入描述】

1行, 包含一个字符串 *str* 。

【输出描述】

1行, 包含最长小写字串的长度。

【样例1】

输入

1 | abcdeACzxc

输出

1 | 5

【概念】

当串 *a* 中连续包含串 *b* 的所有元素时,  
串 *a* 是串 *b* 的父串;  
串 *b* 是串 *a* 的子串。  
例如 *abcdefg* 是 *cde* 的子串。

目录

尺取法

一、复习枚举算法

- 1. 算法三要素
- 2. 最小公倍数公式
- 3. 时间复杂度

二、算法优化初级

- 1. 概念
- 2. 例题
  - (1) 最长小写字串
  - (2) 最长递增子串
  - (3) 最小子串和
  - (4) 最短字符串包含
- II 代码




I 初步算法

```
1 for (int i = 0; i <= len-1; i++)
2 {
3     for (int j = i; j <= len-1; j++)
4     {
5         bool flag = true;
6         for (int k = i; k <= j; k++)
7         {
8             if (a[k] >= 'A' && a[k] <= 'Z')
9             {
10                bool flag = true;
11                break;
12            }
13        }
14        if (flag)
15        {
16            // ...
17        }
18    }
19 }
20 // 打擂台...
21 // 输出...
```

II 认识尺取法

尺取法, 又称 **双指针** 法, 是一种针对子串枚举问题的优化算法。  
在上述题目中, 可用下面的伪代码来表示。

```
1 while (l < len) // 左指针没有到结尾
2 {
3     if ((s[r] >= 'a' && s[r] <= 'z') && r < len) // 右指针在小写字母上并且不在结尾
4     {
5         r++;
6     }
7 }
```

AI Code Thunder

关注

👍 0 🗨 0 ⭐ 0 💬 0

专栏目录

```
10         max(..., r-1)
11         l = r + 1;
12         r = l;
13     }
}
```

目录

尺取法

一、复习枚举算法

- 1. 算法三要素
- 2. 最小公倍数公式
- 3. 时间复杂度

二、算法优化初级

- 1. 概念
- 2. 例题
  - (1) 最长小写字串
  - (2) 最长递增子串
  - (3) 最小子串和
  - (4) 最短字符串包含
- II 代码

III 尺取法程序

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  int main()
6  {
7      // 输入
8      string s;
9      cin >> s;
10
11     // 尺取法
12     int len = s.length();
13     int l = 0, r = 0; // l 表示左端点, r 表示右端点
14     int maxlen = 0;
15     while (l < len)
16     {
17         if (s[r] >= 'a' && s[r] <= 'z' && r < len) // 右指针在小写字母上并且不在结尾
18         {
19             r++;
20         }
21         else
22         {
23             maxlen = max(maxlen, r-l);
24             l = r + 1; // 移动左指针
25             r = l; // 右指针和左指针一起移动
26         }
27     }
28
29     // 输出
30     cout << maxlen;
31     return 0;
32 }
```

(2) 最长递增子串

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  int main()
6  {
7      // 输入
8      int n, s[10005] = {};
9      cin >> n;
10     for (int i = 0; i <= n-1; i++)
11     {
12         cin >> s[i];
13     }
14
15     // 尺取法
16     int l = 0, r = 0, maxlen = 1;
17     while (l < n)
18     {
19         if (s[r] < s[r+1] && r < n)
20         {
21             r++;
22         }
23         else
24         {
```

 AI Code Thunder

关注

 0   0  0 

专栏目录

```
27         r = 1;
28     }
29 }
30
31 // 输出
32 cout << maxlen;
33 return 0;
34 return 0;
35 }
```

(3) 最小子串和

I 伪代码

```
1 while (l < len)
2 {
3     if (sum < 9 && r < len)
4     {
5         sum += a[r];
6         r++;
7     }
8     else
9     {
10        if (sum >= 9)
11            r
```

目录

尺取法

一、复习枚举算法


- 1. 算法三要素
- 2. 最小公倍数公式
- 3. 时间复杂度

二、算法优化初级

- 1. 概念
  - 2. 例题
    - (1) 最长小写字串
    - (2) 最长递增子串
    - (3) 最小子串和
    - (4) 最短字符串包含
- II 代码

II 完整代码

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     // 输入
7     int n, s, a[10005] = {};
8     cin >> n;
9     for (int i = 0; i <= n-1; i++)
10    {
11        cin >> a[i];
12    }
13    cin >> s;
14
15    // 尺取法
16    int l = 0, r = 0, minlen = 1e8, sum = 0;
17    while (l < n)
18    {
19        if (sum < s && r < n)
20        {
21            sum += a[r];
22            r++;
23        }
24        else
25        {
26            if (sum >= s)
27            {
28                minlen = min(minlen, r-l);
29            }
30            sum -= a[l];
31            l++;
32        }
33    }
34 }
```

AI Code Thunder

关注

 0

 0

 0

 0



专栏目录

```
37     {
38         cout << 0;
39     }
40     else
41     {
42         cout << minlen;
43     }
44     return 0;
45 }
```

(4) 最短字符串包含

I 伪代码

```
1 while (l < len)
2 {
3     if (sum <= 26 && r < len) // sum 表示出现了多少个不同的字母
4     {
5         cnt[s[r]]++;
6         if (cnt[s[r]] == 1)
7         {
8             sum++;
9         }
10        r++;
11    }
```

目录

尺取法

一、复习枚举算法

- 1. 算法三要素
- 2. 最小公倍数公式
- 3. 时间复杂度

二、算法优化初级

- 1. 概念
  - 2. 例题
    - (1) 最长小写字串
    - (2) 最长递增子串
    - (3) 最小子串和
    - (4) 最短字符串包含
- II 代码

II 代码

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 int main()
6 {
7     string s;
8     cin >> s;
9
10    int len = s.length();
11    int l = 0, r = 0;
12    int cnt[130] = {};
13    int sum = 0;
14    int minlen = 1e8;
15    while (l < len)
16    {
17        if (sum < 26 && r < len)
18        {
19            cnt[s[r]]++;
20            if (cnt[s[r]] == 1) sum++;
21            r++;
22        }
23        else
24        {
25            if (sum >= 26)
26            {
27                minlen = min(minlen, r-l);
28            }
29            cnt[s[l]]--;
30            if (cnt[s[l]] == 0)
31            {
32                sum--;
33            }
34            l++;
```

复制



AI Code Thunder

关注

 0  0  0  0



专栏目录