

基础算法

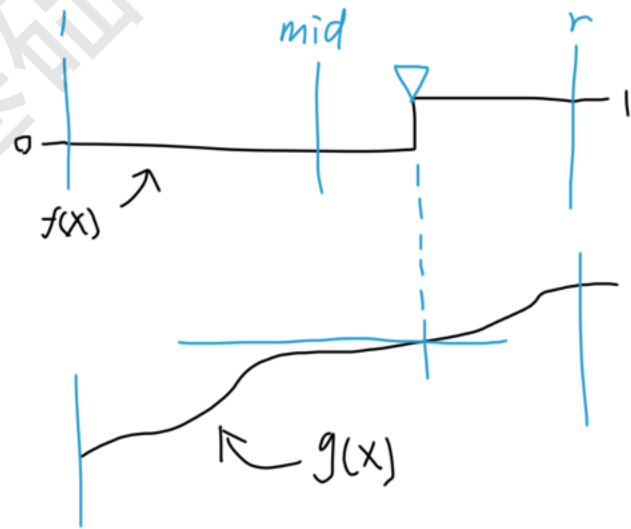
二分

对于一个取值只有01，且只在一个地方突变的函数，我们可以用二分的方式找到它的突变点。

而对于一个单调的函数，它是否小于某个值便是一个满足上述条件的函数。

所以二分的关键点在于，找到并证明一个函数的单调性或突变性，并用合适的复杂度计算它。

二分复杂度 $O(\log n)$



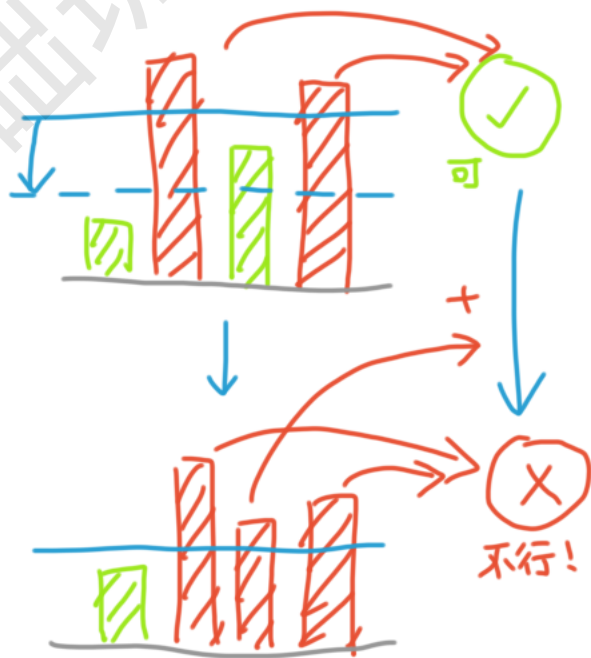
P1525 [NOIP2010 提高组] 关押罪犯

S 城现有两座监狱，一共关押着 N 名罪犯。他们之间的关系自然也就极不和谐。如果客观条件具备则随时可能爆发冲突。我们用“怨气值”（一个正整数值）来表示某两名罪犯之间的仇恨程度，怨气值越大，则这两名罪犯之间的积怨越多。如果两名怨气值为 c 的罪犯被关押在同一监狱，他们俩之间会发生摩擦，并造成影响力为 c 的冲突事件。共有 M 对的不和谐关系。

请你把罪犯分到两座监狱，使影响力最大的冲突事件最小。

$N \leq 20000, M \leq 100000$ 。

我们发现最大值最小或最小值最大经常暗示题目需要使用二分。这是因为（以最大值最小为例），我们使结果更小的代价是限制不断增多，这些限制是大于结果的东西不能出现。而越发增多的限制最终使得某些条件不能成立，使得我们无法一直使结果更小。这里蕴含着两重单调性：1. 限制一旦出现后不再消失 2. 条件在不成立后不会因为限制增多后又变得成立。这使得我们可以通过二分解题。



贪心

对于题目中的某些操作，你有了一个只看眼前，看起来很假的策略，然后你证明出它是真的，那么这个策略就是一个好的贪心。

可以看出，贪心的难点在于如何找到一个策略并证明它是对的。

对于找策略并没有一种普适性的方法，但也有一些套路，如排序法，后悔法。

(所以贪心很玄学)

P1182 数列分段 Section II

对于给定的一个长度为 N 的正整数数列 A ，现要将其分成 M 段，并要求每段连续，且每段和的最大值最小。

$$N \leq 20000$$

广度优先搜索和深度优先搜索

两种搜索方式都可以使程序访问所有可能到达的节点。但访问的顺序不同。

广度优先搜索按照起始节点到此节点所需最小边数从小到大的顺序访问（dis数组便是起始节点到此节点所需最小边数）。

深度优先搜索一直沿着一条出边深入，直到无边可走后退回。访问顺序可形成一个树的结构。

```
1 void dfs(int nod) {
2     if(v[nod]) return;
3     v[nod]=1;
4     ...
5     for(nod.edge)
6         dfs(edge.to);
7 }
8 dfs(begin);
9
10
11 q.push(begin); v[begin]=1; dis[begin]=0;
12 while(q.size()) {
13     int nod=q.front(); q.pop();
14     ...
15     for(nod.edge)
16         if(!v[edge.to]) {
17             v[edge.to]=1;
18             dis[edge.to]=dis[nod]+1;
19             q.push(edge.to);
20         }
21 }
```

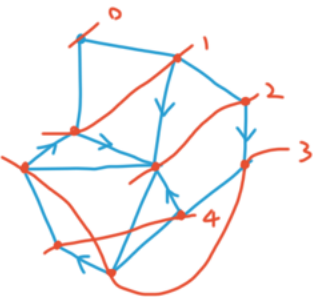
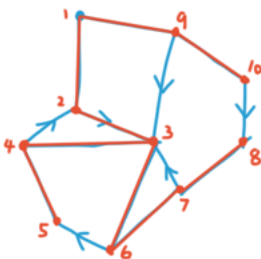
广度优先搜索和深度优先搜索

这两种搜索的访问顺序不同，导致他们有不同的使用场景。

广度优先搜索时常与最短路有关。

深度优先搜索的访问顺序DFS序和欧拉序是解决部分图论问题的有力工具。

同时，点和边可以在题目中隐式地表达。一个状态可以看作点，状态间的转化可以看作边，从而用搜索解决。



P6037 Ryoku 的探索

Ryoku 所处的世界可以抽象成一个有 n 个点， n 条边的带权无向连通图 G 。每条边有美观度和长度。

Ryoku 会使用这样一个策略探索世界：在每个点寻找一个**她未走过的**边中**美观度最高的**走，如果没有边走，就沿着她前往这个点的边返回，类似于图的**深度优先遍历**。

探索的一个方案的长度是这个方案所经过的所有边长度的和（返回时经过的长度不用计算）。

她想知道，对于每一个起点 $s=1,2,\dots,n$ ，她需要走过的长度是多少？

$N \leq 10^6$

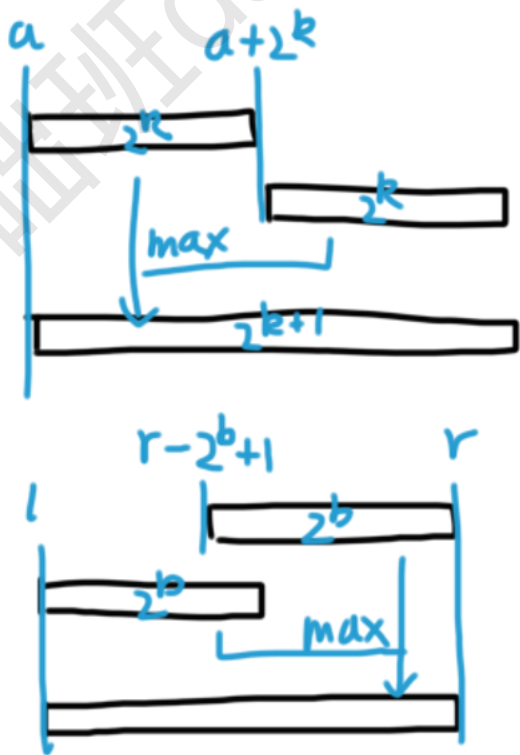
RMQ问题

即多次询问一个序列在区间 $[l, r]$ 的最值得问题。

我们通过倍增预处理出以某个位置起始长度为2的次幂的区间的极值。

然后由于最值运算的可叠加性，我们把询问的区间分为两个长度为2的次幂的部分。用预处理的数据计算即可。

预处理复杂度 $O(n \log n)$ ，询问复杂度 $O(1)$



给定一个序列，多次询问询问一个序列在区间 $[l,r]$ 的按位与和。

给定一个序列，多次询问询问一个序列在区间 $[l,r]$ 的gcd。

同时RMQ问题在图论中有与欧拉序结合的运用。

构造

构造的题目一般要求是输出一种满足条件的数据或方案。一般结果不唯一。而构造的难点在于如何想出对于所有情况都成功的方式来创造数据或方案，并验证它的正确性。

而构造的方法千奇百怪，没有太多的套路。

(所以构造也很玄学)

操作题

你有两个数 a 和 b ，其中 a 等于0， b 等于1，你有2种操作：

操作一：你可以选择 a 的值加上 b 或 b 的值加上 a 。

操作二：你可以选择 a 或 b 乘以数字 x 。

请你输出一个操作次数不超过400次的操作序列，使得 a 或 b 中有一个值等于 n 。

其中 x 和 n 的值为给定的。

$0 \leq n \leq 10^9, 2 \leq x \leq 10$ 。

<https://ac.nowcoder.com/acm/contest/11228/D>

字符串的存储与读取

string类型: cin>>□

char[]类型: gets(□), scanf("%s",□), cin>>□

[a][b][c][_][d][e][f][\n]...

string cin>> [a][b][c]

char[] gets() [a][b][c][_][d][e][f][\0]

char[] scanf("%s",) [a][b][c][\0]

char[] cin>> [a][b][c][\0]

字符串的存储与输出

string类型: cout<<□

char[]类型: puts(□), printf("%s",□), cout<<□

char[]	[<u>a</u>] [<u>b</u>] [<u>c</u>] [<u>_</u>] [<u>d</u>] [<u>e</u>] [<u>f</u>] [<u>\0</u>]
char[] puts()	[<u>a</u>] [<u>b</u>] [<u>c</u>] [<u>_</u>] [<u>d</u>] [<u>e</u>] [<u>f</u>] [<u>\n</u>]
char[] printf("%s",)	[<u>a</u>] [<u>b</u>] [<u>c</u>] [<u>_</u>] [<u>d</u>] [<u>e</u>] [<u>f</u>]
char[] cout>>	[<u>a</u>] [<u>b</u>] [<u>c</u>] [<u>_</u>] [<u>d</u>] [<u>e</u>] [<u>f</u>]
string	[<u>a</u>] [<u>b</u>] [<u>c</u>] [<u>_</u>] [<u>d</u>] [<u>e</u>] [<u>f</u>]
string cout>>	[<u>a</u>] [<u>b</u>] [<u>c</u>] [<u>_</u>] [<u>d</u>] [<u>e</u>] [<u>f</u>]

字符串的遍历与头文件

两种储存方式都可以直接用下标随机访问，但遍历的方式略有不同。

所需头文件：

cin,cout <iostream>

string <string>

scanf,printf <cstdio>

gets,puts <cstdio>

strlen <cstring>

```
1 string s;  
2     for(int i=0;i<s.size();i++)  
3         s[i]...;  
4  
5  
6 char s[...];  
7     for(int i=0;s[i];i++)  
8         s[i]...;  
9  
10    int l=strlen(s);  
11    for(int i=0;i<l;i++)  
12        s[i]...;  
13  
14    #[error,0(l^2)]  
15    for(int i=0;i<strlen(s);i++)  
16        s[i]...;
```


P5015 [NOIP2018 普及组] 标题统计

凯凯刚写了一篇美妙的作文，请问这篇作文的标题中有多少个字符？统计标题字符数时，空格和换行符不计算在内。

对于 100% 的数据， $1 \leq |s| \leq 5$ ，输入可能包含大、小写英文字母、数字字符、空格和行末换行符。

前缀和与差分

对于一个序列 a ，设其前缀和数组为 a^+ ，差分数组为 a^- 。

则有：（下标从1开始）

$$\begin{aligned}a_1^+ &= a_1, a_i^+ = a_{i-1}^+ + a_i \\ a_1^- &= a_1, a_i^- = a_i - a_{i-1}\end{aligned}$$

可以发现一些性质：

$$\begin{aligned}(a^+)^- &= (a^-)^+ = a \\ (a+b)^+ &= a^+ + b^+, (a+b)^- = a^- + b^-\end{aligned}$$

前缀和与差分

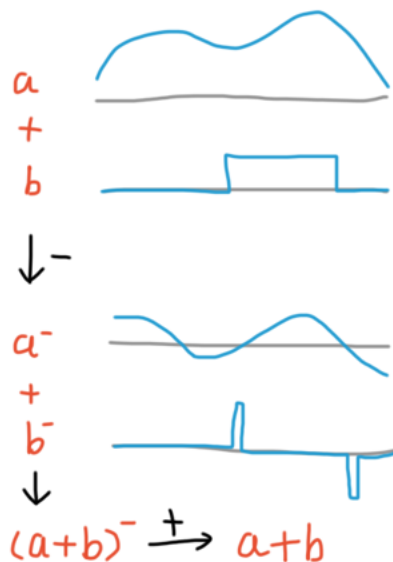
前缀和能让我们快速地计算序列中某一段的和。

$[l, r]$ 这段中的和为 $a_r^+ - a_{l-1}^+$ 。

差分能让我们把连续加一段相同的数的操作转化为两点的修改。

(根据前页的性质发现其成立)。

此技巧可在树状数组中使用。



P1360 [USACO07MAR]Gold Balanced Lineup G

有 m 个数，初始为全0。共有 n 天，每天对某些给定的位置加1。请你求出一个最长的时间段，使得在这个时间段内每个数增加的大小相同。输出此长度。

$m \leq 30, n \leq 10^5$

P7962 [NOIP2021] 方差

给定长度为 n 的非严格递增正整数数列 $1 \leq a_1 \leq a_2 \leq \dots \leq a_n$ 。每次可以进行的操作是：任意选择一个正整数 $1 < i < n$ ，将 a_i 变为 $a_{i-1} + a_{i+1} - a_i$ 。求在若干次操作之后，该数列的方差最小值是多少。请输出最小值乘以 n^2 的结果。

离散化

有着将值域过大的数据缩小的能力，
并且不改变数据间的大小关系。

分为预处理和查询两个部分。

预处理对全体数据排序并去重，复
杂度 $O(n\log n)$

查询是通过二分得知离散化的结果，
单次复杂度 $O(\log n)$

此代码需使用<algorithm>

```
1  sort(a+1,a+tot+1);  
2  int l=unique(a+1,a+tot+1)-a-1;  
3  
4  v_=lower_bound(a+1,a+len+1,v)-a;
```

有一无限长的数列，初始为有部分位置有值，位置记为 pos 。有 m 次操作，每次对 $[l,r]$ 范围内的位置加1，问 m 次操作后有多少位置的数大于 k ？

$m \leq 10^5, pos \leq 10^{18}, l \leq r \leq 10^{18}$

分治

分治一般分为几步：1.将原问题分为几个部分 2.将这几个部分作为新问题进行递归地解决 3.将这几个部分的结果合并并处理这几个部分间的关系对整体问题的影响（如果能够这么做的话）

这一思想可以用在多个结构上：序列，平面空间，树

P1115 最大子段和

给出一个长度为 n 的序列 a ，选出其中连续且非空的一段使得这段和最大。

$$n \leq 2 \times 10^5$$

<https://blog.csdn.net/TheWayForDream/article/details/116759814>

P1966 [NOIP2013 提高组] 火柴排队 (有修改)

涵涵有两盒火柴，每盒装有 n 根火柴，每根火柴都有一个高度。现在将每盒中的火柴各自排成一列，同一列火柴的高度互不相同，两列火柴之间的距离定义为： $\sum_i (a_i - b_i)^2$

每列火柴中相邻两根火柴的位置都可以交换，请你通过交换使得两列火柴之间的距离最小。请求出这个最小的距离。

$$n \leq 10^5$$

P2949 [USACO09OPEN]Work Scheduling G

约翰有 n 个工作，每项工作花一个单位时间。他的工作日从 0 时刻开始。在任一时刻，他都可以选择 N 项工作中的任意一项工作来完成。每个单位时间里只能做一个工作，而每项工作又有一个截止日期 D_i ，如果他可以完成这个工作，那么他可以获利 P_i 。在给定的工作利润和截止时间下，约翰能够获得的利润最大为多少。

$$n \leq 10^5$$