

随机化算法

cssyz - yzy

- 本人对于随机化算法**由衷的喜爱**（当年捆绑数据下随机数拿10pts
- 浅显的讲解随机化算法的基础原理（我也只会基础
- 配套相应的例题加深理解（其正解往往不是随机化，看阳寿

目录：

- 0. 关于随机数
- 1. 随机函数
- 2. 爬山问题
- 3. 模拟退火
- 4. 例题

关于随机数

- 随机数与伪随机数

你能在 OI 界看到的随机数均为**伪随机数**。

现有的计算机的运算过程具有确定性，违反了随机化的不可预测的性质。

真正的随机数并不好用，物极必反。

随机函数： `rand()`

- 可用于生成伪随机数。
- 使用 `rand()` 函数时需要调用一个 `srand(seed)` 函数，名为“播种函数”（我取的。
- 其中 `seed` 为种子。通常情况下，种子取 `time(0)`，既系统时间函数。
- ps：想想为什么。
- 显然，因为在同一个电脑，同一个编译器上，取用同一个随机种子调用 `rand()` 函数的结果相同。

如何用 `rand()` 生成 `long long` 值域下的随机数

- 众所周知，`rand()` 生成的随机非负整数值域为 $[0, \text{RAND_MAX}]$ ，其中在 Linux 系统下 $\text{RAND_MAX} = 2^{31} - 1$ ，在 Windows 系统下 $\text{RAND_MAX} = 2^{15} - 1$ 。
- ps：思考，如何用 `rand()` 生成 `long long` 值域下的随机数。
- 利用拼接，将两个 `int` 范围内的随机数通过位运算拼接在一起。
- `(rand() << 32 | rand())` 用于生成 `long long` 范围内的整数。

随机函数 `mt19937`

- **梅森旋转算法** 是一个伪随机数发生算法，基于 32 位梅森缠绕器。由松本真和西村拓士在1997年开发;
- 梅森旋转的变体 `mt19937` 可以产生32位整数序列;
- 一个有着 19937 位状态大小的能够生成 32 位数的梅森旋转伪随机生成器
- **状态大小**: 指的是在随机数生成器中用于存储和更新内部状态的位数或字节数。其影响着生成的随机数序列的质量和性能。
- 周期长，速度快，固定维度均等分布;

随机函数 `mt19937_64`

- 基于 64 位梅森缠绕器;
- 使用方式同 `mt19937` , 但随机数范围扩大到了 `unsigned long long` 类型的取值范围。

随机函数 `random_shuffle()`

- 随机打乱指定序列;
- 性能缺失，不均匀;
- `random_shuffle(l, r)` 左闭右开;
- 来自 OI-wiki：`random_shuffle()` 已于 C++14 标准中被弃用，于 C++17 标准中被移除。

爬山算法

爬山算法是一种启发式搜索方法，用于解决优化问题。

- 类似与搜索算法，不断寻找最优解;
- 引入温度 t 的概念，温度越高，解的搜索范围越大;
- 每次寻找最有解，不会有概率的选择差解;

ps：会有什么问题，如何优化？

我们只要想办法让差解也能在某时刻被接受，从而跳出局部最有解。

这样我们就将爬山算法优化成了**模拟退火**算法。

模拟退火

为什么要称为模拟退火？

- 因为它在模拟退火;
- 由于退火的规律引入了更多随机因素，那么我们得到最优解的概率会大大增加;

- 对于爬山算法，我们只要再加上一个选择非最优解的概率 $P(A)$ 即可。
- 类次爬山算法：引入初温 T_0 ，降温系数 dt ，末温 T_n ，不断从 T_0 降温至 T_n 。
- 温度越大，状态差值越大，更希望浪；
- 设两次状态之差为 ΔE ，则选择概率为：

$$P(\Delta E) = \begin{cases} 1, & S' \text{ is better than } S, \\ e^{\frac{-|\Delta E|}{T}}, & \text{otherwise.} \end{cases}$$

- 其中状态的变化量和 T 决定概率 P 的大小，所以这个值是动态的。

动态流程

From OI-wiki

算法流程

口头叙述（懒

Algorithm 1 模拟退火算法

```
1:  $x \leftarrow x_0$ 
2:  $T \leftarrow T_0$ 
3:  $k \leftarrow 0$ 
4: while  $k \leq k_{max}$  and  $T \geq T_f$  do
5:    $x_k \leftarrow \text{NEIGHBOR}(s)$ 
6:    $\Delta f = f(x_k) - f(x)$ 
7:   if  $\Delta f < 0$  or  $\text{RANDOM}(0, 1) \leq P(\Delta f, T)$  then
8:      $x \leftarrow x_k$ 
9:   end if
10:   $T \leftarrow \text{COOLING}(T, k, k_{max})$ 
11:   $k \leftarrow k + 1$ 
12: end while
```

例题

luoguP1284 三角形牧场

有 n 块木板，长度为 l_i 。将所有的木板任意拼接后，拼成一个三角形。

最大化三角形面积 S 。

$$3 \leq n \leq 40, 1 \leq l_i \leq 40$$

luoguP4035 「JSOI2008」球形空间产生器

给出 n 维空间中的 $n + 1$ 个点，已知它们在同一个 n 维球面上，求出球心。

$n \leq 10$ ，坐标绝对值不超过 20000。

luoguP3878 [TJOI2010] 分金币

T 组数据。

有 n 枚金币，价值为 v_i 。

把它们分成两部分，要求这两部分金币数目之差不超过 1。

最小化两部分金币的价值之差。

$1 \leq T \leq 20$, $1 \leq n \leq 30$, $1 \leq v_i \leq 2^{30}$ 。

luoguP1337 [JSOI2004] 平衡点 / 吊打XXX

有 n 个重物，重量 w_i ，坐标 x_i, y_i ，每个重物系在一条足够长的绳子上。

求绳结 x 最终平衡于何处. (最小化 $\sum_{i=1}^n w_i \times dis_{i,x}$)

Thank you for watching