循环不变式证明：

第i次迭代前怎样，然后怎样操作的，之后i+1次迭代前还是这样

时间复杂度

- 循环的复杂度是i的取值个数1？
- 循环内每行，要考虑这是在循环里

比较asymptotic

- 相加类，看最大幂次
- lgn比n任意正次幂小
- 不直观，一定取对数看看
- 由定义

$$\lg^*(\lg n) = \lg^*(n) - 1$$

- 由斯特朗公式，n！=n^n

$$
\begin{aligned}
1 = n^{1/\lg n} &\ll \\
\lg(\lg^* n) &\ll \\
\boxed{\lg^* n = \lg^*(\lg n)} &\ll \\
2^{\lg^* n} &\ll \\
\ln \ln n &\ll \\
\sqrt{\lg n} &\ll \\
\ln n &\ll \\
\lg^2 n &\ll \\
2^{\sqrt{2 \lg n}} &\ll \\
(\sqrt{2})^{\lg n} &\ll \\
n = 2^{\lg n} &\ll \\
\boxed{n \lg n = \lg(n!)} &\ll \\
n^2 = 4^{\lg n} &\ll \\
n^3 &\ll \\
(\lg n)! &\ll \\
(\lg n)^{\lg n} = n^{\lg \lg n} &\ll \\
(3/2)^n &\ll \\
2^n &\ll \\
n \cdot 2^n &\ll \\
e^n &\ll \\
n! &\ll \\
(n+1)! &\ll \\
2^{2^n} &\ll \\
2^{2^{n+1}} &
\end{aligned}
$$

递归树时间计算

定理 4.1（主定理） 令 $a \geqslant 1$ 和 $b > 1$ 是常数，$f(n)$ 是一个函数，$T(n)$ 是定义在非负整数上的递归式：

$$T(n) = aT(n/b) + f(n)$$

其中我们将 n/b 解释为 $\lfloor n/b \rfloor$ 或 $\lceil n/b \rceil$。那么 $T(n)$ 有如下渐近界：

1. 若对某个常数 $\varepsilon > 0$ 有 $f(n) = O(n^{\log_b a - \varepsilon})$，则 $T(n) = \Theta(n^{\log_b a})$。
2. 若 $f(n) = \Theta(n^{\log_b a})$，则 $T(n) = \Theta(n^{\log_b a} \lg n)$。
3. 若对某个常数 $\varepsilon > 0$ 有 $f(n) = \Omega(n^{\log_b a + \varepsilon})$，且对某个常数 $c < 1$ 和所有足够大的 n 有 $af(n/b) \leqslant cf(n)$，则 $T(n) = \Theta(f(n))$。 ■

```
T(n) = T(α) + T(n - α) + cn
```

```
    常数切割Θ(nn)
T(n) = T(αn) + T((1 - α)n) + cn
    常数倍切割Θ(nlgn)
```

讲作业：

对顶堆：最大堆+中位数+最小堆，要维护两个堆数量的平衡

可删除堆：其实没删，只是又创建了一个要删的堆，每次访问最大时要进行判断，如果是要删的就跳过

lowerbound只能用决策树证明，如果叶子总数易知，那就好证