

给定正整数 n 。

求 $n!$ 的结尾有多少个0。

人生第一道算法题，第一次都是痛并快乐着。

刚学了oop，并没有overflow，algorithm之类的概念。

导致第一版的解法是直接开了个int存结果。

$0 \leq N \leq 10^{18}$

然后int连 $n = 20$ 都没撑住。

debug半天才知道还能overflow。遂上网查了半天，孩子成功用上了long。

并且坚持到了 $n = 30$ 。（然并卵

```
// 最后发现 只要有 2 和 5 就能得到一个结尾0。因为  $2 * 5 = 10$ 。
```

```
// (10, 15, 20...同理)
```

```
// 因为  $2 * 10 = 2 * 2 * 5$ 
```

```
// 任意一个 5的倍数都可以拆分成  $b * 5$ 。
```

```
//  $2 * b * 5 = b0$ ;
```

```
// 所以说判断 【0, N】一共有多少5即可。
```

```
//  $O(n)$  会超时。
```

```
// 但是判断5可以快快的判断。
```

```
// 最优为 $O(\log(n))$ 
```

举个栗子：

判断 $100!$ 有多少个结尾0。

显然 2 的个数大于 5，因为 $2 \dots 5 \dots 12 \dots 15$ 所以说你遇到一个5一定会先遇到一个2。

注意到 25 有两个 5。

1-100 有几个五呢？（15 --》 $3*5$ ||| $a * 15 = a * 5 * 3$ ，所以说10 15 20 都算一个5）

$100 / 5 = 20$;

$20 / 5 = 4$;

$4 / 5 = 0$;

答案为 $20 + 4$. 24个0结尾。

当时十一点写到半夜四点才写出来。。。

刚刚在图书馆只用了40秒。/(T o T)/