شایان کبریایی ۴۰۰۲۴۳۰۹۵ پیچیدگی زمانی مینی پروژه ساختمان داده ها

کد اول: تعداد بارهایی که if(i%j==0) اجرا شود به طور حدودی :

$$1+2+\cdots+n-2=\frac{(n-1)(n-2)}{2}\in \underline{O(n^2)}$$

کد دوم: مثل قسمت قبلی :

$$(\sqrt{4}-1)+(\sqrt{5}-1)+\cdots+(\sqrt{n}-1)$$

$$=(\sqrt{4}+\sqrt{5}+\cdots+\sqrt{n})-(n-3)$$

$$\approx \frac{(\sqrt{n}+\sqrt{4})(n-3)}{2}-(n-3)=\frac{(\sqrt{n}+\sqrt{4}-2)(n-3)}{2}$$

به طور حدودی

$$\in \underline{O(n\sqrt{n})}$$

کد سوم: تعداد بارهایی که hasDivisor[i]=true اجرا شود به طور حدودی:

$$\frac{n}{2}+\frac{n}{3}+\cdots+1=n\sum_{i=2}^{n}\frac{1}{i}\approx n\left(1+\frac{\lg n}{2}\right)\longrightarrow 2^k=n\longrightarrow k=\lg n$$

طبق سری هارمونیک

بنابراین کل برنامه به این میزان operation دارد:

$$n+n\left(1+\frac{\lg n}{2}\right)+n\approx 2n+n\lg n\in \underline{O(n\lg n)}\in O(n\lg n)$$

مقداردهی اولیه آرایه

hasDivisor.count(True)

کد چهارم ، مثل کد قبلی :

کد نهایتاً تا ($i = \sqrt{n}$) اجرا می‌شود و بعد از آن به هیچ وجه وارد حلقه دومی نمی‌شود. (علت آن نیز واضح است : $\sqrt{n}\times\sqrt{n}=n$)

بنابراین :

$$\frac{n}{2}+\frac{n}{3}+\dots+\frac{n}{\sqrt{n}}$$

$$= n\sum_{i=2}^{\sqrt{n}} \geq n\left(1+\frac{\lg\sqrt{n}}{2}\right) \longrightarrow 2^{k}=\sqrt{n} \to k=\lg\sqrt{n}$$

طبق سری هارمونیک

اولیه آرایه مقداردهی

const.

$$n + n\left(1+\frac{\lg\sqrt{n}}{2}\right) + n =$$

$$3n + \frac{n\lg\sqrt{n}}{2} \in O(n\lg\sqrt{n})$$

$$\in O(n\log\sqrt{n})$$

$$\in O(n\log n)$$