

class Parent {

Parent() {

System.out.println("Parent constructor");

}

class Child extends Parent {

Child() {

print("Child constructor");

}

}

class Test {

main() {

Child c = new Child();

}

}

Given a & b
find the count of primes \leq
 a & b

$$[1 \leq a \leq 5 \leq 10^7]$$

$$\begin{aligned} & \log_{10} 10^7 = 7 \\ & \log_2 7 \approx 2.8 \\ & 2.8 \times 10^7 \\ & 32 \times 10^7 \\ & = 3.2 \times 10^9 \end{aligned}$$

$$a = 1, \quad \delta = 7$$

$$10^7 \sqrt{10^7}$$

$$10^7 \times 10^3 \times \sqrt{10}$$

$$\underline{10^{10} \times \sqrt{10}}$$

$$1 \text{ to } N$$

$$N \times \lg \lg(N)$$

LR
Meth 2: Power

Recursion 2

Power

A, B, C

$$(A^B) \cdot 1.C$$

fast Power

$$(A \times B) \cdot 1.C$$

$$= ((A \cdot 1.C) \times (B \cdot 1.C)) \cdot 1.C$$

$$A^B \cdot 1.C = \left((A^{B/2}) \cdot 1.C \right) \times (A^{B/2}) \cdot 1.C$$

$$\left. \begin{array}{l} -10^9 \leq A \leq 10^9 \\ 0 \leq B \leq 10^9 \\ 0 \leq C \leq 10^9 \end{array} \right\} \text{Old}$$

$$A \times (A^{B/2}) \times (A^{B/2})$$

Assumption

$$\text{power} \Rightarrow [0, 10^9-1]$$

int power (A, B, C) {
if (A == 0) return
if (B == 0) {

Returns $A^B \cdot 1.C$
0 {
↓
[0, C-1]

return 1;

int p = power(A, B/2, C) ~~· C~~ ⇒ [0, 10⁹-1]

if (B % 2 == 0) {

return ((long)(p * p) % C)

} else {

return ((long)(p * p) % C * (long)A) % C;

{0, 10⁹}

p²⁷

22nd Friday

25th Monday H₁ H₂ H₃ H₄ X

27th Wed

29th Fri

} Off

3rd ~~Monday~~ ^{Wednesday} \Rightarrow Class