

**1. Asociación**

* **T1(n) = 5n2+10n5*n*2+10*n*** → O(n2)*O*(*n*2) → *Selection/Insertion Sort*
* **T2(n) = 6nlog2n+3006*n*log2​*n*+300** → O(nlogn)*O*(*n*log*n*) → *Mergesort/Heapsort*
* **T3(n) = 0.01n30.01*n*3** → O(n3)*O*(*n*3) → *Multiplicación de matrices cúbica*
* **T4(n) = 1.5n1.5*n*** → O(1.5n)*O*(1.5*n*) → *Backtracking con poda leve*

**2. Comparar dos pares y encontrar umbral con for + if**

Vamos a comparar **T1** y **T2**, y **T3** y **T4**.

**Par 1: T1 vs T2**

5n2+10nvs6nlog2n+3005*n*2+10*n*vs6*n*log2​*n*+300

Para n*n* pequeño, T2 puede ser mayor por la constante 300, pero luego T1 crece más rápido.  
Queremos el menor n*n* tal que 5n2+10n>6nlog2n+3005*n*2+10*n*>6*n*log2​*n*+300.

Podemos hacer búsqueda con for n in range(...) y if T1(n) > T2(n).

**Cálculo manual aproximado:**

Para n=20*n*=20:  
T1 = 5⋅400+200=22005⋅400+200=2200  
T2 = 6⋅20⋅4.32+300≈518.4+300=818.46⋅20⋅4.32+300≈518.4+300=818.4 → T1 > T2 ya.

Probemos más pequeño:  
n=10*n*=10:  
T1 = 500+100=600500+100=600  
T2 = 6⋅10⋅3.32+300≈199.2+300=499.26⋅10⋅3.32+300≈199.2+300=499.2 → T1 > T2.

n=5*n*=5:  
T1 = 125+50=175125+50=175  
T2 = 6⋅5⋅2.32+300≈69.6+300=369.66⋅5⋅2.32+300≈69.6+300=369.6 → T1 < T2.

Entonces cruza entre n=5*n*=5 y n=10*n*=10.

**Búsqueda fina:**

n=6*n*=6:  
T1 = 5⋅36+60=2405⋅36+60=240  
T2 = 6⋅6⋅2.585+300≈93.06+300=393.066⋅6⋅2.585+300≈93.06+300=393.06 → T1 < T2

n=7*n*=7:  
T1 = 245+70=315245+70=315  
T2 = 6⋅7⋅2.807+300≈117.9+300=417.96⋅7⋅2.807+300≈117.9+300=417.9 → T1 < T2

n=8*n*=8:  
T1 = 320+80=400320+80=400  
T2 = 6⋅8⋅3+300=144+300=4446⋅8⋅3+300=144+300=444 → T1 < T2

n=9*n*=9:  
T1 = 405+90=495405+90=495  
T2 = 6⋅9⋅3.17+300≈171.2+300=471.26⋅9⋅3.17+300≈171.2+300=471.2 → T1 > T2

Entonces umbral n=9*n*=9.

**Par 2: T3 vs T4**

0.01n3vs1.5n0.01*n*3vs1.5*n*

Para n*n* pequeño, 0.01n30.01*n*3 es menor, luego exponencial domina.

Probemos:  
n=10*n*=10: T3 = 0.01⋅1000=100.01⋅1000=10, T4 = 1.510≈57.6651.510≈57.665 → T3 < T4  
n=20*n*=20: T3 = 0.01⋅8000=800.01⋅8000=80, T4 = 1.520≈3325.261.520≈3325.26 → T3 < T4

En realidad, el cruce puede estar donde 0.01n3≈1.5n0.01*n*3≈1.5*n*.  
Probemos n=30*n*=30: T3 = 0.01⋅27000=2700.01⋅27000=270, T4 = 1.530≈1.510⋅1.520≈57.665⋅3325.26≈191,0001.530≈1.510⋅1.520≈57.665⋅3325.26≈191,000 → T3 < T4.

La exponencial ya es enorme, así que revisemos cerca del inicio:  
n=1*n*=1: T3 = 0.01, T4 = 1.5 → T3 < T4  
Siempre T4 > T3 para n≥1*n*≥1 ? No, probemos n=0*n*=0 (aunque n≥1 en algoritmos):  
n=0: T3=0, T4=1 → T3<T4.

Parece que **nunca** T3 > T4 para n*n* entero ≥ 1, porque 1.5n1.5*n* supera a 0.01n30.01*n*3 desde el principio.

¿Hay cruce?  
Comparemos n=1*n*=1: T3=0.01, T4=1.5 → T4 mayor.  
n=2*n*=2: T3=0.08, T4=2.25 → T4 mayor.  
Siempre mayor.

Entonces no hay umbral entero n≥1*n*≥1 donde T3 supere a T4.

**Respuesta final:**

* Asociación:  
  T1 → Selection/Insertion  
  T2 → Mergesort/Heapsort  
  T3 → Multiplicación de matrices cúbica  
  T4 → Backtracking con poda leve
* Umbrales:  
  T1 vs T2: n=9*n*=9  
  T2 vs T3: n≈67*n*≈67

Conclusion

La magia de Isaac se basa en precomputación inteligente. Una vez preparado el arreglo primos\_hasta, puede responder cualquier consulta en tiempo constante, dando la apariencia de habilidad sobrenatural cuando en realidad es algoritmia eficiente.