Problemas P y NP

Existe un problema abierto en la informática en el que no se sabe a ciencia cierta si los problemas de complejidad P son iguales a los problemas de complejidad NP, los problemas P son polinomiales esto significa que el tiempo de ejecución se puede representar como un polinomio estos problemas se pueden resolver un numero definido de pasos ya que es decidible mientras que los problemas NP son los problemas que al contrario de P su solución es mucho mas tardada ya que no es polinomial resulta que la complejidad NP es exponencial.

Entonces aquí el problema es que normalmente en un problema P se puede resolver fácilmente y aparte se puede comprobar que el resultado está bien muy rápido como por ejemplo al sacar la raíz cuadrada de un numero solo basta multiplicar el resultado por si mismo para saber si el resultado fue correcto

```c
#include <math.h>
#include <stdio.h>
int main(int argc, char const *argv[])
{
    double n;
    printf("calculadora de raices cuadradas\n");
    printf("Selecciona un numero\n");
    scanf("%lf", &n);
    double resultado = sqrt(n);
    printf("El resultado es %lf\n", resultado);
    printf("\n");
    printf("Comprobando\n");
    double comprobacion = resultado * resultado ;
    printf("Comprobacion: %lf * %lf =", resultado, resultado);
    printf(" %lf es igual al numero original %lf\n", comprobacion, n);
    return 0;
}
```

Pero en un programa NP que no es polinomial comprobar el resultado es muy sencillo, pero llegar a este resultado es algo que se puede llegar a complicar demasiado debido al crecimiento exponencial, esto pasa en programas que resuelven puzles como un rompecabezas o un sudoku. Entonces la cuestión es que se esta buscando si es posible que en un problema NP que normalmente se comprueba fácilmente haya la posibilidad de que se pueda llegar a la solución de forma sencilla y con un numero finito de pasos como ocurre en los problemas de clase P