

Complexidade de Algoritmos

Paulino Ng

2020-02-26

Ementa

- ▶ Complexidade e desempenho.
- ▶ Análise da complexidade.
- ▶ Medidas de Complexidade.
- ▶ Comparação entre algoritmos recursivos e iterativos.
- ▶ Complexidade em algoritmos de busca e ordenação.
- ▶ Classes de problemas P, NP, NP-completo e NP- difícil.
- ▶ Estratégias para projetar algoritmos.
- ▶ Métodos de redução de problemas.

Complexidade de um programa simples

- ▶ Nesta e nas próximas transparências, vamos usar um C anterior ao C ANSI
- ▶ Seja o programa simples abaixo:

```
main()  
{  
    printf("hello, world\n");  
}
```

Modificado: Último teorema de Fermat: $x^n + y^n = z^n$

```
int exp(int i, n) {
    int ans, j;  ans = 1;
    for (j=1; j<=n; j++) ans *= i;
    return(ans);
}

main() {
    int n, total, x, y, z;
    scanf("%d", &n);  total = 3;
    while (1) {
        for (x=1; x<=total-2; x++)
            for (y=1; y<=total-x-1; y++) {
                z = total - x - y;
                if (exp(x,n) + exp(y,n) == exp(z,n))
                    printf("hello, world\n");
            }
        total++;
    }
}
```

Questão

O programa modificado vai imprimir “hello, world” como seus 12 primeiros caracteres?

Resposta

- ▶ Para $n = 2$, ele vai imprimir diversas vezes `hello, world`
- ▶ Para $n > 2$, o programa entra em loop infinito sem imprimir nada
 - ▶ Os matemáticos levaram mais de 300 anos para concluir que o teorema de Fermat, $x^n + y^n = z^n$, para x, y, z, n naturais só tem solução para $n = 2$ estava certo.

Como saber se um programa termina?