

Convite à Análise de Algoritmos

**Bacharelado em Ciência da Computação
Universidade Federal Rural de Pernambuco**

Rodrigo de Souza

15 de Agosto de 2011

Um problema lógico

Problema SAT

Instância Fórmula lógica φ :

- variáveis booleanas x_1, \dots, x_n ;
- conectivos lógicos \wedge , \vee e \neg ;
- valores lógicos V/F

Pergunta Existe uma atribuição de valores lógicos V/F a x_1, \dots, x_n tal que φ resulte em V?

Um problema lógico

Problema SAT

Instância Fórmula lógica φ :

- variáveis booleanas x_1, \dots, x_n ;
- conectivos lógicos \wedge , \vee e \neg ;
- valores lógicos V/F

Pergunta Existe uma atribuição de valores lógicos V/F a x_1, \dots, x_n tal que φ resulte em V?

$$\varphi = (\neg x_1 \wedge x_2) \vee (x_2 \wedge x_3)$$

φ é V para (FVV)

φ é F para (VVF)

Um algoritmo para SAT: ENUM

- Gere $(V \dots VV), (V \dots VF), (V \dots FV), \dots, (F \dots FF)$
- Se encontrou sequência tal que φ é V, responda SIM e pare
- Responda NÃO

Um algoritmo para SAT: ENUM

- Gere $(V \dots VV), (V \dots VF), (V \dots FV), \dots, (F \dots FF)$
- Se encontrou sequência tal que φ é V, responda SIM e pare
- Responda NÃO

ENUM é bom?

Um algoritmo para SAT: ENUM

- Gere $(V \dots VV), (V \dots VF), (V \dots FV), \dots, (F \dots FF)$
- Se encontrou sequência tal que φ é V, responda SIM e pare
- Responda NÃO

ENUM é bom?

ENUM é eficiente?

Um algoritmo para SAT: ENUM

- Gere $(V \dots VV), (V \dots VF), (V \dots FV), \dots, (F \dots FF)$
- Se encontrou sequência tal que φ é V, responda SIM e pare
- Responda NÃO

ENUM é bom?

ENUM é eficiente?

Como medir eficiência...

... sem computador/cronômetro?

Um algoritmo para SAT: ENUM

- Gere $(V \dots VV), (V \dots VF), (V \dots FV), \dots, (F \dots FF)$
- Se encontrou sequência tal que φ é V, responda SIM e pare
- Responda NÃO

sequências geradas (*no pior caso*)

$$2^n$$

Um algoritmo para SAT: ENUM

- Gere $(V \dots VV), (V \dots VF), (V \dots FV), \dots, (F \dots FF)$
- Se encontrou sequência tal que φ é V, responda SIM e pare
- Responda NÃO

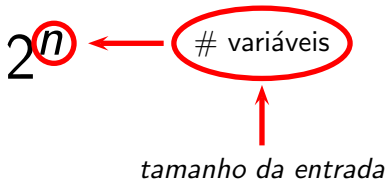
sequências geradas (*no pior caso*)

$$2^n \longleftarrow \# \text{ variáveis}$$

Um algoritmo para SAT: ENUM

- Gere $(V \dots VV), (V \dots VF), (V \dots FV), \dots, (F \dots FF)$
- Se encontrou sequência tal que φ é V, responda SIM e pare
- Responda NÃO

sequências geradas (*no pior caso*)



Um algoritmo para SAT: ENUM

- Gere $(V \dots VV), (V \dots VF), (V \dots FV), \dots, (F \dots FF)$
- Se encontrou sequência tal que φ é V, responda SIM e pare
- Responda NÃO

sequências geradas (*no pior caso*)

$$2^n$$

\sim # operações de ENUM em função de n (no pior caso)

Um algoritmo para SAT: ENUM

- Gere $(V \dots VV), (V \dots VF), (V \dots FV), \dots, (F \dots FF)$
- Se encontrou sequência tal que φ é V, responda SIM e pare
- Responda NÃO

sequências geradas (*no pior caso*)

$$2^n$$

\sim # operações de ENUM em função de n (*no pior caso*)

complexidade de ENUM (*no pior caso*)

Um algoritmo para SAT: ENUM

- Gere $(V \dots VV), (V \dots VF), (V \dots FV), \dots, (F \dots FF)$
- Se encontrou sequência tal que φ é V, responda SIM e pare
- Responda NÃO

sequências geradas (*no pior caso*)

$$2^n$$

\sim # operações de ENUM em função de n (*no pior caso*)

complexidade de ENUM (*no pior caso*)

boa medida da eficiência de ENUM

Um algoritmo para SAT: ENUM

- Gere $(V \dots VV), (V \dots VF), (V \dots FV), \dots, (F \dots FF)$
- Se encontrou sequência tal que φ é V, responda SIM e pare
- Responda NÃO

sequências geradas (*no pior caso*)

$$2^n$$

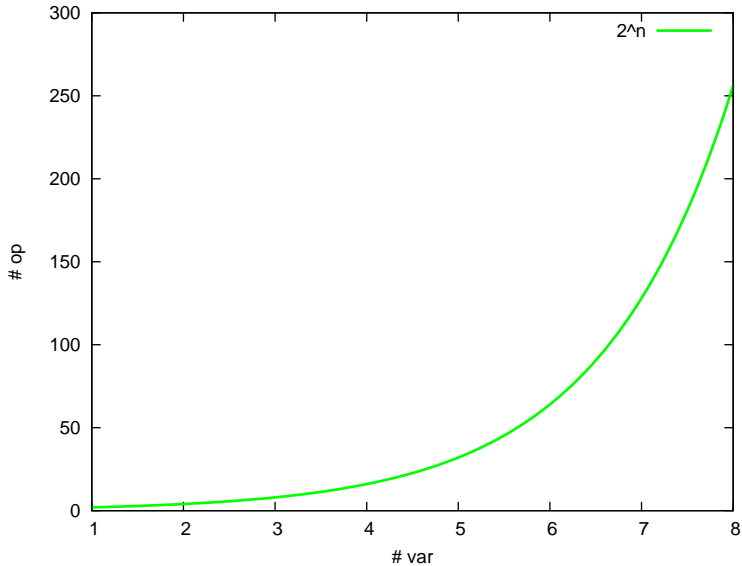
\sim # operações de ENUM em função de n (*no pior caso*)

complexidade de ENUM (*no pior caso*)

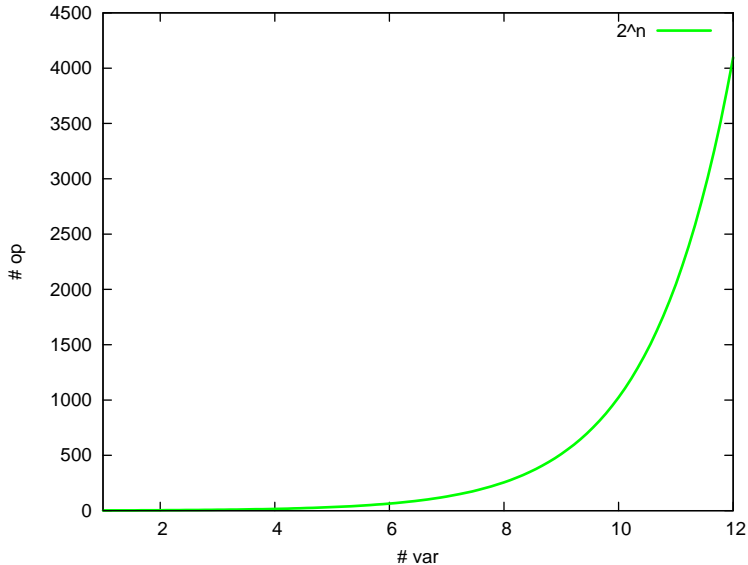
boa medida da eficiência de ENUM

ENUM é eficiente?

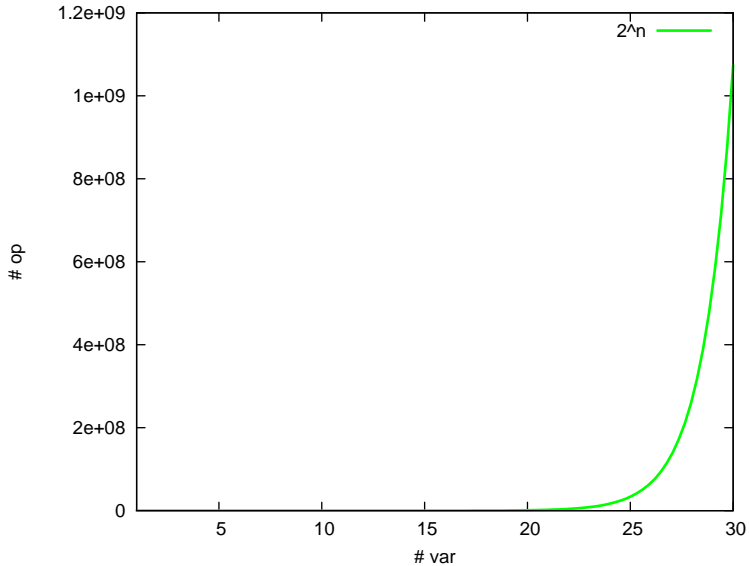
operações ENUM, até 8 variáveis



operações ENUM, até 12 variáveis



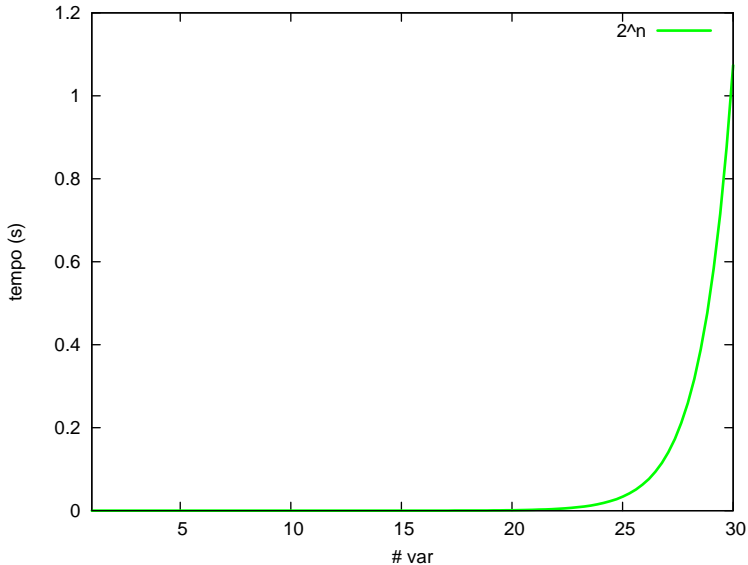
operações ENUM, até 30 variáveis



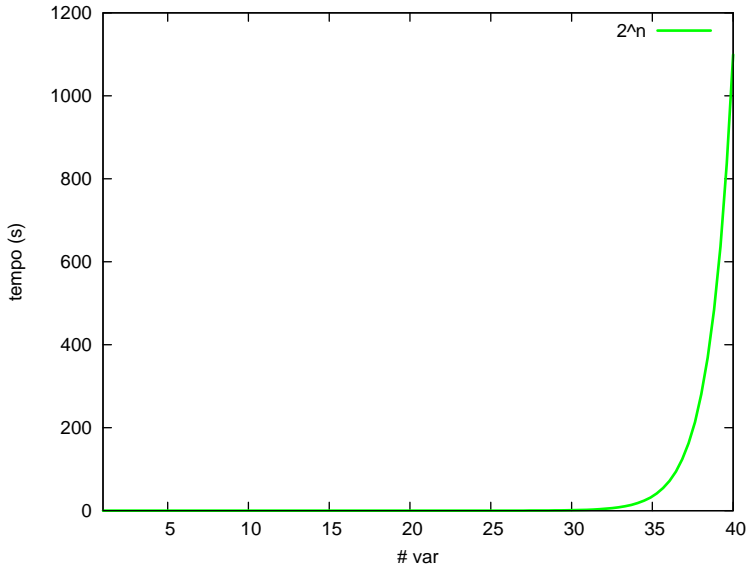
Processador FASTA

10^9 operações
por segundo

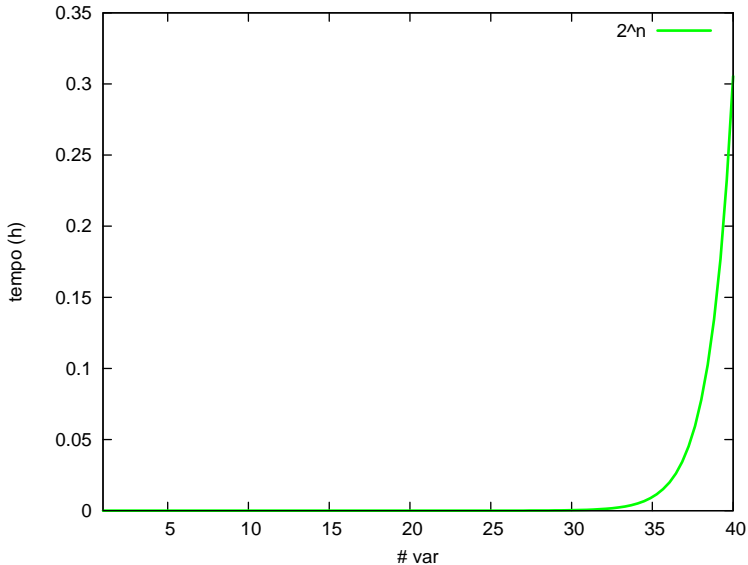
Tempo em s ENUM, FASTA, até 30 variáveis



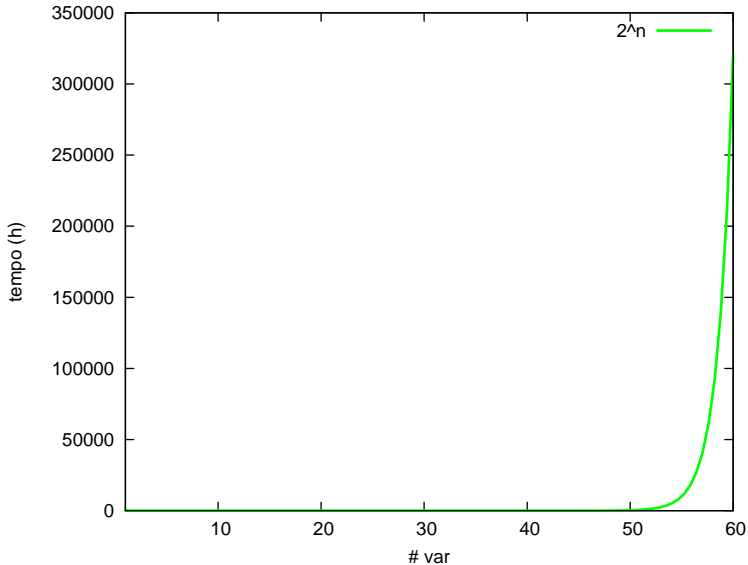
Tempo em s ENUM, FASTA, até 40 variáveis



Tempo em h ENUM, FASTA, até 40 variáveis



Tempo em h ENUM, FASTA, até 60 variáveis

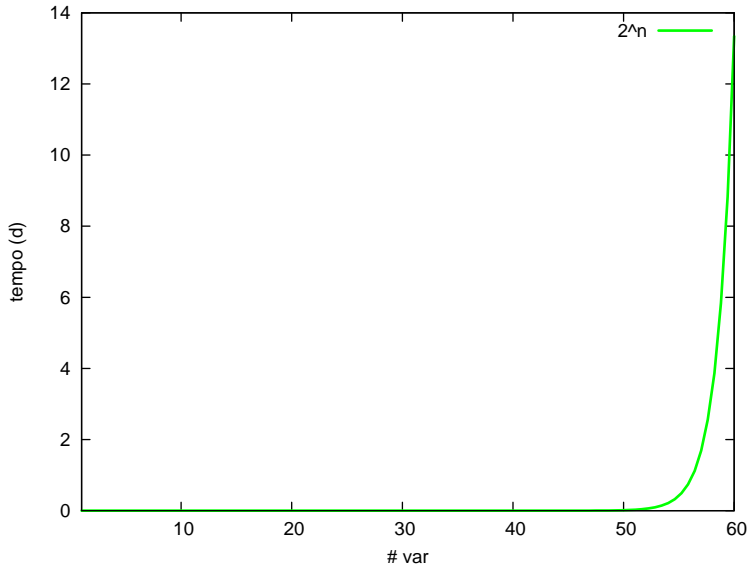


Processador FASTA-M

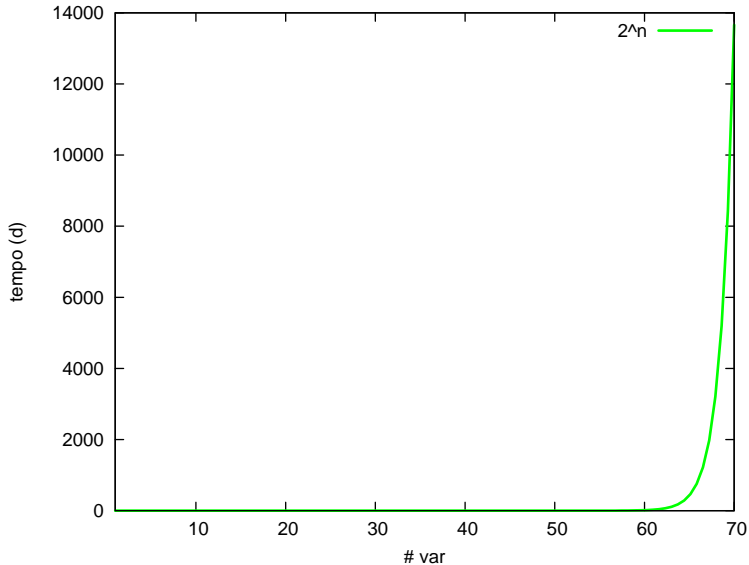
10^{12} operações
por segundo

1000 vezes mais rápido que
FASTA

Tempo em d ENUM, FASTA-M, até 60 variáveis



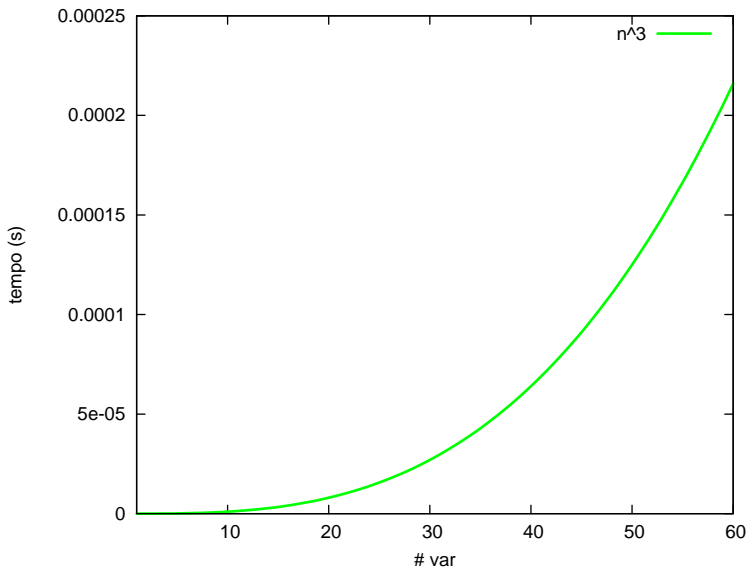
Tempo em d ENUM, FASTA-M, até 70 variáveis



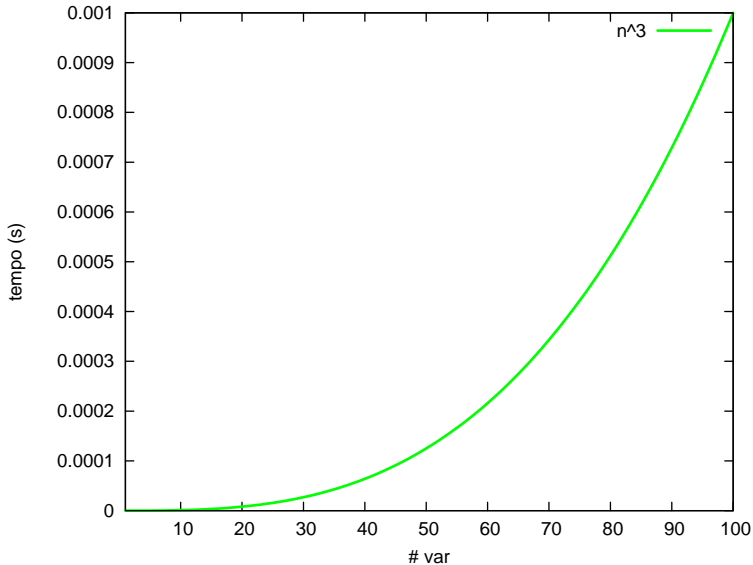
Algoritmo HIPO

Resolve SAT
em n^3 operações

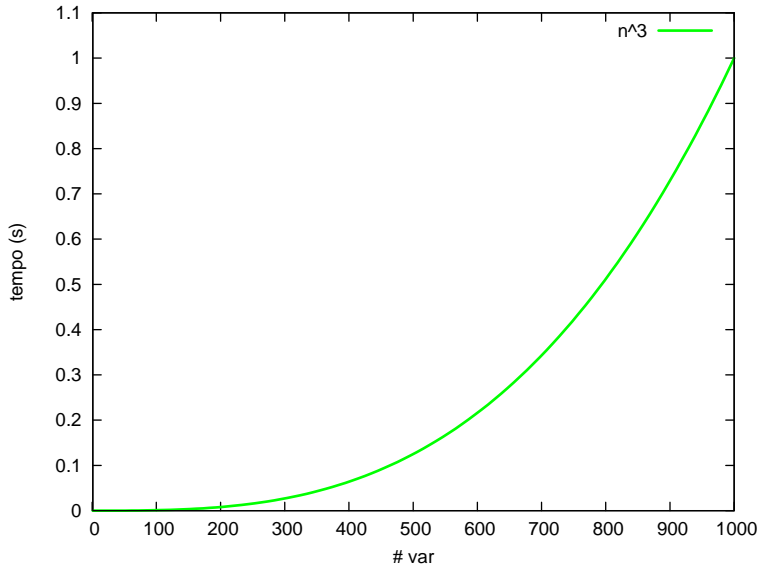
Tempo em s HIPO, FASTA, até 60 variáveis



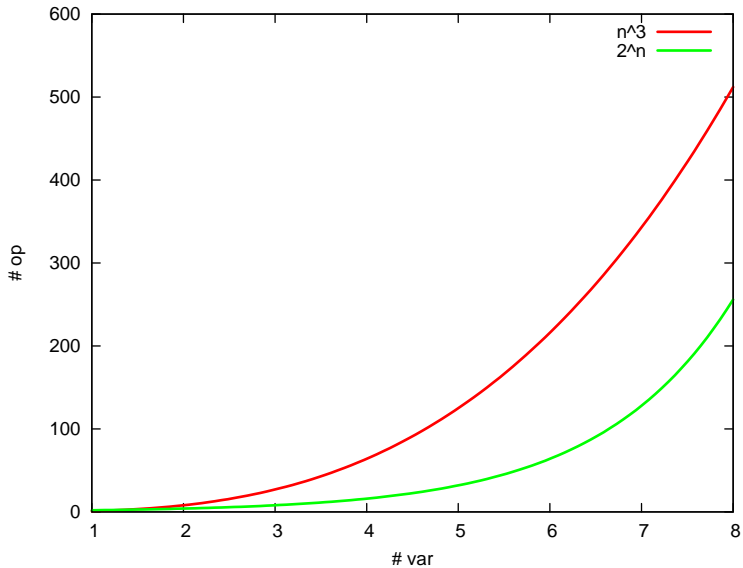
Tempo em s HIPO, FASTA, até 100 variáveis



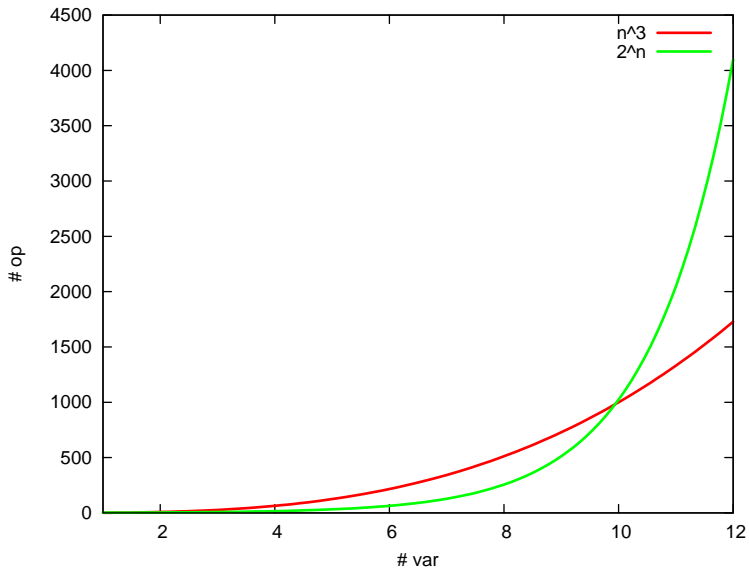
Tempo em s HIPO, FASTA, até 1000 variáveis



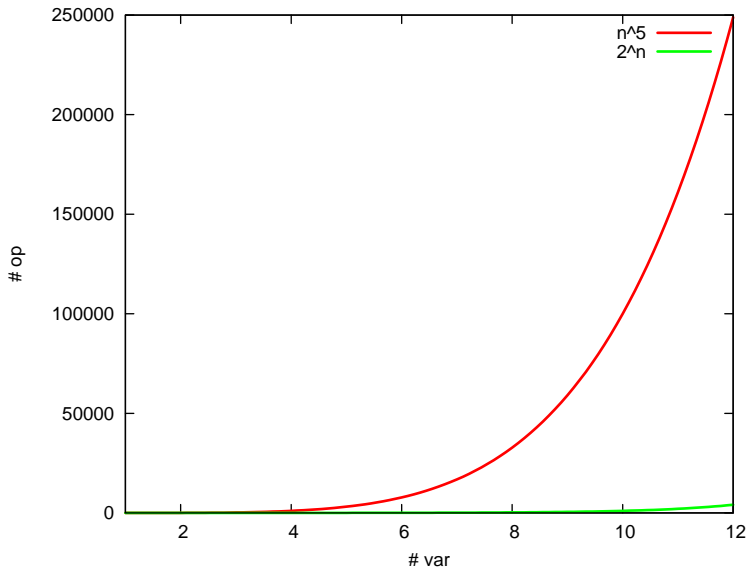
operações ENUM/HIPO, até 8 variáveis



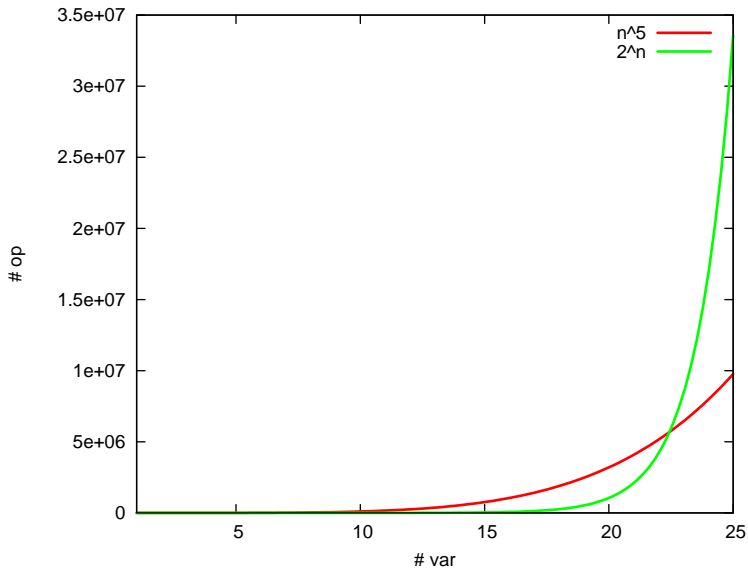
operações ENUM/HIPO, até 12 variáveis



operações ENUM/HIPO-SLOW, até 12 variáveis



operações ENUM/HIPO-SLOW, até 25 variáveis



Constatações

- melhor tecnologia: solução temporária
- melhor algoritmo: solução definitiva

Constatações

- melhor tecnologia: solução temporária
- melhor algoritmo: solução definitiva

- complexidade: medida matemática de eficiência
- valores pequenos da entrada não importam
- complexidade polinomial: razoável
- complexidade exponencial: impraticável

Constatações

- melhor tecnologia: solução temporária
- melhor algoritmo: solução definitiva
- complexidade: medida matemática de eficiência
- valores pequenos da entrada não importam
- complexidade polinomial: razoável
- complexidade exponencial: impraticável

Tarefa: encontre HIP0

Constatações

- melhor tecnologia: solução temporária
- melhor algoritmo: solução definitiva
- complexidade: medida matemática de eficiência
- valores pequenos da entrada não importam
- complexidade polinomial: razoável
- complexidade exponencial: impraticável

Tarefa: encontre HIPO

(brincadeira...)

Novas questões

- Que operações contar em um algoritmo?
- Como comparar algoritmos?
- Como encontrar um algoritmos mais eficiente?
- Como saber se meu algoritmo é ótimo?

Novas questões

- Que operações contar em um algoritmo?
As essenciais, que “dominam” as demais
- Como comparar algoritmos?
- Como encontrar um algoritmos mais eficiente?
- Como saber se meu algoritmo é ótimo?

Novas questões

- Que operações contar em um algoritmo?
As essenciais, que “dominam” as demais
- Como comparar algoritmos?
Ignore constantes, compare crescimento de complexidades
- Como encontrar um algoritmos mais eficiente?
- Como saber se meu algoritmo é ótimo?

Novas questões

- Que operações contar em um algoritmo?
As essenciais, que “dominam” as demais
- Como comparar algoritmos?
Ignore constantes, compare crescimento de complexidades
- Como encontrar um algoritmos mais eficiente?
Estruturas de dados, propriedades, astúcia...
- Como saber se meu algoritmo é ótimo?

Novas questões

- Que operações contar em um algoritmo?
As essenciais, que “dominam” as demais
- Como comparar algoritmos?
Ignore constantes, compare crescimento de complexidades
- Como encontrar um algoritmos mais eficiente?
Estruturas de dados, propriedades, astúcia...
- Como saber se meu algoritmo é ótimo?
Cota inferior (desconhecida para muitos problemas)

Exercício

Para cada período de tempo e cada função $f(n)$, determine o maior tamanho de instância, n , de um problema quando executado por um algoritmo que requer $f(n)$ microssegundos ($10^{-6}s$)

| | 1 segundo | 1 minuto | 1 hora | 1 dia | 1 mês | 1 ano | 1 século |
|------------|--------------|-------------|-----------|----------|----------|----------|-------------|
| $\lg n$ | | | | | | | |
| \sqrt{n} | | | | | | | |
| n | | | | | | | |
| $n \lg n$ | | | | | | | |
| n^2 | | | | | | | |
| n^3 | | | | | | | |
| 2^n | | | | | | | |
| $n!$ | | | | | | | |