Universidade Federal do Ceará
Departamento de Computação

Construção e Análise de Algoritmos
Trabalho de Implementação 1

Relembre o Algoritmo 1 da Questão 4 da Lista 1, descrito abaixo:

Questão 4 da Lista 1 Considere o algoritmo NumRecursivo(N) abaixo. Determine o valor retornado em notação assintótica.

Algorithm 1 inteiro NumRecursivo(inteiro N)

```
Valor = 0     //(Valor é uma variável local)
if N > 1 then
   Valor = Valor + NumRecursivo(⌊N/5⌋) + NumRecursivo(⌊N/5⌋)
   Valor = Valor + NumRecursivo(⌊N/6⌋) + NumRecursivo(⌊N/6⌋) + NumRecursivo(⌊N/6⌋)
   Valor = Valor + NumRecursivo(⌊N/10⌋) + N
end if
retorne Valor
```

(i) Implemente esse algoritmo na linguagem de sua preferência e monte uma curva com pelo menos 1000 valores retornados desse algoritmo para valores de N de 100.000 em 100.000.
Ou seja, $N = 10^5,\ 2 \cdot 10^5,\ 3 \cdot 10^5,\ 4 \cdot 10^5,\ \ldots$.

(ii) Na Lista 1, é pedido para provar que o número retornado pelo algoritmo é da ordem assintótica de $\Theta(n \cdot \log(n))$. Com os dados da curva que você obteve no parágrafo anterior, tente estimar qual seria a melhor base B do logaritmo de modo que o número retornado seja aproximadamente $\approx n \log_B(n)$.
(**Dica:** Obter uma curva de bases e tirar a média).

(iii) Finalmente, tente fazer alguma conta teórica (sem usar as curvas geradas) que dê uma boa estimativa para essa base B do logaritmo e compare com o valor obtido no parágrafo anterior.
(**Dica:** indução matemática ignorando casos bases).

Repita esse procedimento anterior para os seguintes Algoritmos 2 a 7:

Algorithm 2 inteiro NumRecursivo(inteiro N)

```
Valor = 0     //(Valor é uma variável local)
if N > 1 then
   Valor = Valor + NumRecursivo(⌊N/5⌋) + NumRecursivo(⌊N/5⌋) + NumRecursivo(⌊N/5⌋)
   Valor = Valor + NumRecursivo(⌊N/6⌋) + NumRecursivo(⌊N/6⌋)
   Valor = Valor + NumRecursivo(⌊N/15⌋) + N
end if
retorne Valor
```

Algorithm 3 inteiro NUMRECURSIVO(inteiro N)

Valor = 0 //(Valor é uma variável local)
if $N > 1$ **then**
  Valor = Valor + NUMRECURSIVO($\lfloor N/5 \rfloor$) + NUMRECURSIVO($\lfloor N/5 \rfloor$)
  Valor = Valor + NUMRECURSIVO($\lfloor N/5 \rfloor$) + NUMRECURSIVO($\lfloor N/5 \rfloor$)
  Valor = Valor + NUMRECURSIVO($\lfloor N/6 \rfloor$) + NUMRECURSIVO($\lfloor N/30 \rfloor$) + N
end if
retorne Valor

Algorithm 4 inteiro NUMRECURSIVO(inteiro N)

Valor = 0 //(Valor é uma variável local)
if $N > 1$ **then**
  Valor = Valor + NUMRECURSIVO($\lfloor N/5 \rfloor$)
  Valor = Valor + NUMRECURSIVO($\lfloor N/6 \rfloor$) + NUMRECURSIVO($\lfloor N/6 \rfloor$)
  Valor = Valor + NUMRECURSIVO($\lfloor N/6 \rfloor$) + NUMRECURSIVO($\lfloor N/6 \rfloor$)
  Valor = Valor + NUMRECURSIVO($\lfloor N/15 \rfloor$) + NUMRECURSIVO($\lfloor N/15 \rfloor$) + N
end if
retorne Valor

Algorithm 5 inteiro NUMRECURSIVO(inteiro N)

Valor = 0 //(Valor é uma variável local)
if $N > 1$ **then**
  Valor = Valor + NUMRECURSIVO($\lfloor N/6 \rfloor$) + NUMRECURSIVO($\lfloor N/6 \rfloor$) + NUMRECURSIVO($\lfloor N/6 \rfloor$)
  Valor = Valor + NUMRECURSIVO($\lfloor N/6 \rfloor$) + NUMRECURSIVO($\lfloor N/6 \rfloor$)
  Valor = Valor + NUMRECURSIVO($\lfloor N/12 \rfloor$) + NUMRECURSIVO($\lfloor N/12 \rfloor$) + N
end if
retorne Valor

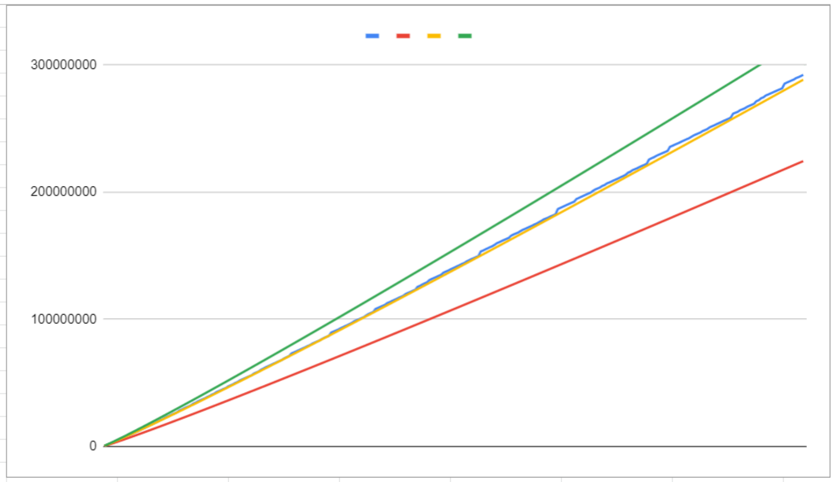
Algorithm 6 inteiro NUMRECURSIVO(inteiro N)

Valor = 0 //(Valor é uma variável local)
if $N > 1$ **then**
  Valor = Valor + NUMRECURSIVO($\lfloor N/5 \rfloor$) + NUMRECURSIVO($\lfloor N/5 \rfloor$) + NUMRECURSIVO($\lfloor N/5 \rfloor$)
  Valor = Valor + NUMRECURSIVO($\lfloor N/5 \rfloor$) + NUMRECURSIVO($\lfloor N/5 \rfloor$) + N
end if
retorne Valor

Algorithm 7 inteiro NUMRECURSIVO(inteiro N)

Valor = 0 //(Valor é uma variável local)
if $N > 1$ **then**
  Valor = Valor + NUMRECURSIVO($\lfloor N/6 \rfloor$) + NUMRECURSIVO($\lfloor N/6 \rfloor$) + NUMRECURSIVO($\lfloor N/6 \rfloor$)
  Valor = Valor + NUMRECURSIVO($\lfloor N/6 \rfloor$) + NUMRECURSIVO($\lfloor N/6 \rfloor$) + NUMRECURSIVO($\lfloor N/6 \rfloor$)
  Valor = Valor + N
end if
retorne Valor

N	T(N)	N*log(N,10)	N*log(N,6)	N*log(N,5)
100000	655040	500000	642548	715338
200000	1372804	1060205	1362467	1516811
300000	2119842	1643136	2111589	2350796
400000	2922151	2240823	2879676	3205894
500000	3719181	2849485	3661865	4076691
600000	4537142	3466890	4455291	4959999
700000	5365392	4091568	5258063	5853711
800000	6150303	4722471	6068835	6756329
900000	6984774	5358818	6886602	7666735
1000000	7806650	5999999	7710583	8584059
1100000	8599873	6645531	8540154	9507606
1200000	9450647	7295017	9374806	10436810
1300000	10265972	7948126	10214114	11371198
1400000	11179151	8604579	11057720	12310369
1500000	11947371	9264136	11905316	13253983
1600000	12721611	9926591	12756635	14201742
1700000	13763697	10591763	13611445	15153387
1800000	14588953	11259490	14469539	16108689
1900000	15381493	11929631	15330736	17067444
2000000	16424093	12602059	16194872	18029471



N	T(N) / N	log(N,10)	log(N,6)	log(N,5)
100000	6.550	5.000	6.425	7.153
200000	6.864	5.301	6.812	7.584
300000	7.066	5.477	7.039	7.836
400000	7.305	5.602	7.199	8.015
500000	7.438	5.699	7.324	8.153
600000	7.562	5.778	7.425	8.267
700000	7.665	5.845	7.512	8.362
800000	7.688	5.903	7.586	8.445
900000	7.761	5.954	7.652	8.519
1000000	7.807	6.000	7.711	8.584
1100000	7.818	6.041	7.764	8.643
1200000	7.876	6.079	7.812	8.697
1300000	7.897	6.114	7.857	8.747
1400000	7.985	6.146	7.898	8.793
1500000	7.965	6.176	7.937	8.836
1600000	7.951	6.204	7.973	8.876
1700000	8.096	6.230	8.007	8.914
1800000	8.105	6.255	8.039	8.949
1900000	8.096	6.279	8.069	8.983
2000000	8.212	6.301	8.097	9.015

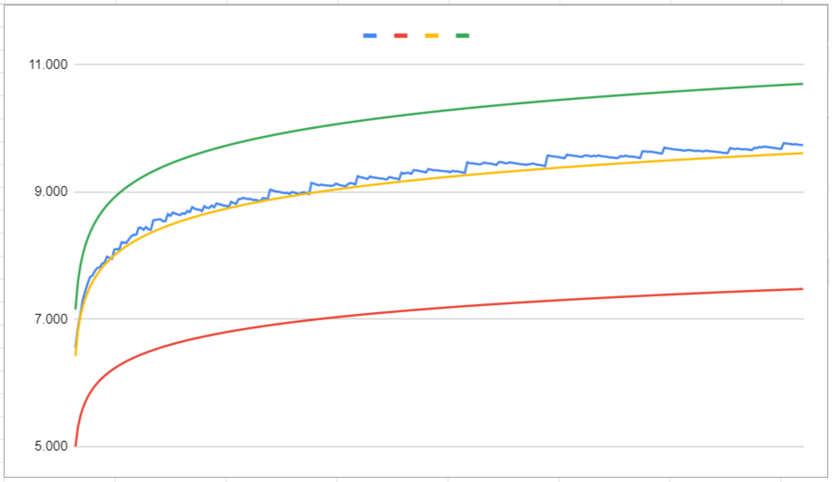


Figura 1: Exemplos de curvas a serem geradas

1 Solução do item (iii) do Algoritmo 1

Faremos a estimativa para a melhor base do logaritmo através de uma indução sem caso base:
Fixe N inteiro grande e seja $T(N)$ o tempo do Algoritmo 1. Suponha que $T(N') \approx N' \cdot \log_B N'$ para todo $N' < N$. Queremos provar que também vale para N, ou seja, $T(N) \approx N \cdot \log_B N$. Vamos tentar estimar o valor de B para que isso ocorra.

Do Algoritmo 1, temos que:

$$T(N) = \quad 2 \cdot T(N/5) \quad + \quad 3 \cdot T(N/6) \quad + \quad T(N/10) \quad + \quad N$$

Pela hipótese da indução, temos que:

$$\begin{aligned}
T(N) &\approx \quad 2 \cdot \left(\frac{N}{5} \log_B \frac{N}{5}\right) \quad + \quad 3 \cdot \left(\frac{N}{6} \log_B \frac{N}{6}\right) \quad + \quad \left(\frac{N}{10} \log_B \frac{N}{10}\right) \quad + \quad N \\
T(N) &\approx \frac{2N}{5}\Big(\log_B N - \log_B 5\Big) + \frac{N}{2}\Big(\log_B N - \log_B 6\Big) + \frac{N}{10}\Big(\log_B N - \log_B 10\Big) + N \\
T(N) &\approx \left(\frac{2N}{5} \log_B N + \frac{N}{2} \log_B N + \frac{N}{10} \log_B N\right) + \left(N - \frac{2N}{5} \log_B 5 - \frac{N}{2} \log_B 6 - \frac{N}{10} \log_B 10\right) \\
T(N) &\approx \left(\frac{2}{5} + \frac{1}{2} + \frac{1}{10}\right) \cdot N \log_B N \quad + \quad N \cdot \left(1 - \frac{2}{5} \log_B 5 - \frac{1}{2} \log_B 6 - \frac{1}{10} \log_B 10\right) \\
T(N) &\approx \left(\frac{4}{10} + \frac{5}{10} + \frac{1}{10}\right) \cdot N \log_B N \quad + \quad N \cdot \left(1 - \log_B 5^{2/5} - \log_B 6^{1/2} - \log_B 10^{1/10}\right) \\
T(N) &\approx \qquad N \cdot \log_B N \quad + \quad N \cdot \left(1 - \log_B \left(5^{2/5} \cdot 6^{1/2} \cdot 10^{1/10}\right)\right)
\end{aligned}$$

Portanto, para que $T(N) \approx N \cdot \log_B N$, temos que $B = 5^{2/5} \cdot 6^{3/6} \cdot 10^{1/10}$.

Portanto $B = 5.870345$.

Observação: Note que essa não é uma conta oficial de indução matemática, visto que não tem caso base e a aproximação $\approx$ não está bem definida. Além disso, a hipótese de indução é muito forte, forçando igualdade $T(N') \approx N' \cdot \log_B N'$ para todo $N' < N$, o que não ocorre (basta verificar os exemplos nos gráficos). Então, formalmente falando, essa conta nem está correta. Mas, na prática, ela funciona sim e podemos considerar o B como sendo mesmo a base teórica, pois podemos provar por indução matemática 100% correta seguindo a mesma estrutura dessa conta que, para N suficientemente grande, $T(N) < N \cdot \log_b N$ para base $b < B$ e $T(N) > N \cdot \log_b N$ para base $b > B$. Resumindo, esta conta não está formalmente correta, mas nos retorna o que queremos de fato: a base B teórica. Veja na figura seguinte como a curva $T(N)/N$ adere bem ao gráfico do logaritmo na base B.

N	(T(N) / N) - log(N,B)	log(N,6) - log(N,B)				
100000	0.046	-0.079				
200000	-0.032	-0.084				
300000	-0.059	-0.087				
400000	0.017	-0.089				
500000	0.024	-0.090				
600000	0.045	-0.092				
700000	0.061	-0.093				
800000	0.008	-0.094				
900000	0.015	-0.094				
1000000	0.001	-0.095				
1100000	-0.042	-0.096				
1200000	-0.033	-0.096				
1300000	-0.057	-0.097				
1400000	-0.011	-0.097				
1500000	-0.070	-0.098				
1600000	-0.120	-0.098				
1700000	-0.009	-0.099				
1800000	-0.033	-0.099				
1900000	-0.073	-0.100				
2000000	0.015	-0.100				
2100000	-0.024	-0.100	0.000			
2200000	-0.047	-0.101	0.000	BASE TEO	B =	5.87034502

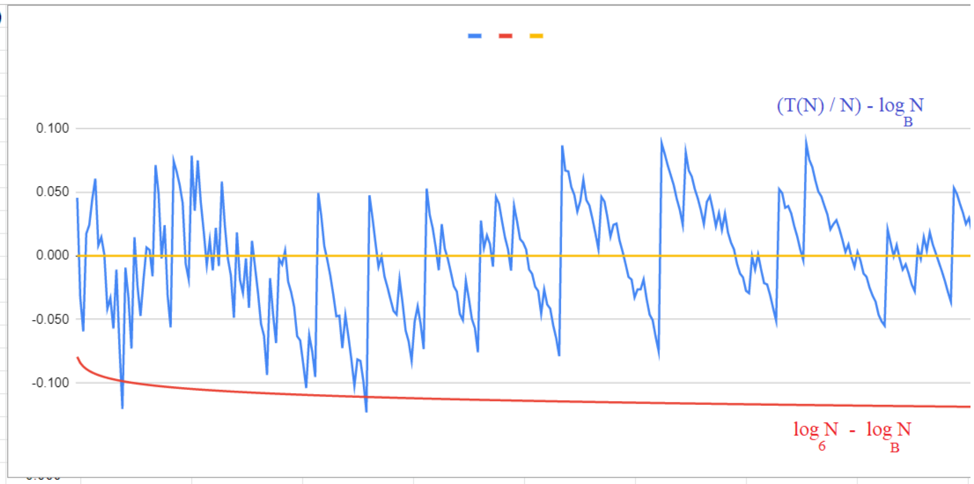


Figura 2: Outro exemplo de curva a ser gerada (agora comparando com a base B teórica)