

Fast Factorials

H.R.Junior e Bruno D. Yamada

23 de Novembro de 2015

1 Intro

Nesses experimentos foram testados dois algoritmos diferentes, um com métodos recursivos (baseados em *divide & conquer*) e outro com um loop definido, de multiplicações sucessivas. Devido a ordem de grandeza dos números envolvidos, fomos obrigados a utilizar também a biblioteca BigInteger, do pacote Java.

Os algoritmos tem muitas semelhanças, como por exemplo a quantidade de operações executadas. A princípio, o tempo de execução de cada uma das abordagens deveria ser similar. Entretanto, como apresentado a seguir, divide and conquer provou ser muito mais rápido do que o método convencional para números muito grandes. A hipótese é que a máquina virtual Java faz operações de multiplicação mais rápido entre dois números de ordem de grandeza similar, ao invés de números com ordens de grandeza discrepantes.

2 Análises

Obtivemos os seguintes dados:



N	FastAlg.	ObviousAlg.
2e0	0,00	0,00
2e1	0,00	0,00
2e2	0,00	0,00
2e3	0,00	0,00
2e4	0,00	0,00
2e5	0,00	0,00
2e6	0,00	0,00
2e7	0,01	0,00
2e8	0,01	0,00
2e9	0,01	0,00
2e10	0,01	0,01
2e11	0,04	0,01
2e12	0,02	0,05
2e13	0,04	0,22
2e14	0,13	0,93
2e15	0,40	4,01
2e16	1,21	17,22
2e17	3,76	73,56
2e18	11,22	323,34
2e19	35,91	1298,39
2e20	103,81	

Como podemos observar, o algoritmo de *Divide&Conquer* (FastAlg.) realiza as operações mais rapidamente do que o algoritmo de loops (ObviousAlg.) para números grandes. A partir de 2e19, o ObviousAlg. toma tempos exageradamente longos. Consta também que o tempo de execução do FastAlg é maior para N entre 2e7 e 2e9. Houveram tentativas de equacionar os tempos de cálculo dos algoritmos, que não obtiveram resultados pois diferentes instâncias do programa geravam diferentes tempos. Houve uma alta taxa de variação de dados, da ordem de aproximadamente 30%.

O arquivo com o número de $10^6!$ (que possui 5.3Mb) foi analisado para a contagem do *longest run of 9's*, e foi constatado que a maior quantidade de algarismos 9 seguidos foram seis.