Except for the number field sieve, all of the asymptotically fast general factoring algorithms have conjectured running times of the above form with $C = 1 + \epsilon$ for ϵ arbitrarily small.

Implications for RSA. Recall that the security of the RSA public key cryptosystem (see § IV.2) depends upon the circumstance that factoring a very large integer of the form $n = pq$ is much more time consuming than the various tasks which legitimate users of the system must perform, tasks which are polynomial time or near-polynomial time (primality testing) as functions of the number r of bits in n. We have just seen why time estimates of the form $O(e^{C\sqrt{r\,log\,r}})$ tend to arise when analyzing factoring algorithms. Since a polynomial function of r can be written in the form $O(e^{C\,log\,r})$, we see that for large r the time required for factorization is indeed much larger than for polynomial time or near-polynomial time algorithms. (However, the factoring algorithms with time estimate of the form $O(e^{C\sqrt{r\,log\,r}})$ are better for large r than the rho method, which has time estimate approximately $O(\sqrt[4]{n}) = O(e^{Cr})$, where $C = \frac{1}{4}\,log\,2$.)

Finally, we note that the question of replacing $\sqrt{r\,log\,r}$ in the exponent by a smaller function of r is not the only matter of practical importance in evaluating the security of the RSA system. After all, a polynomial function of the number of bits r becomes much smaller than $C_1 e^{C_2\sqrt{r\,log\,r}}$ only when r is large, and how large r must be taken depends strongly on the values of the constants C_1 and C_2. So even the discovery of a factoring algorithm with the same time estimate except with smaller constants would have practical implications for the usability of the RSA public key cryptosystem.

Exercises

1. Use Fermat factorization to factor: (a) 8633, (b) 809009, (c) 92296873, (d) 88169891, (e) 4601.
2. Prove that, if n has a factor that is within $\sqrt[4]{n}$ of $\sqrt{n}$, then Fermat factorization works on the first try (i.e., for $t = [\sqrt{n}] + 1$).
3. (a) Prove that if $k = 2$, or if k is any integer divisible by 2 but not by 4, then we cannot factor a large odd integer n using generalized Fermat factorization with this choice of k.
   (b) Prove that if $k = 4$, and if generalized Fermat factorization works for a certain t, then simple Fermat factorization (with $k = 1$) would have worked equally well.
4. Use generalized Fermat factorization to factor: (a) 68987, (b) 29895581, (c) 19578079, (d) 17018759.
5. Let $n = 2701$. Use the B-numbers 52^2, 53^2 *mod* n for a suitable factor-base B to factor 2701. What are the $\vec{\epsilon}$'s corresponding to 52 and 53?
6. Let $n = 4633$. Use 68, 152 and 153 with a suitable factor-base B to factor 4633. What are the corresponding vectors?