Prime Number Generation

George E. Collins

September 23, 1998

SACLIB utilizes three lists of prime numbers: one of small primes (SPRIME), one of medium primes (MPRIME), and one of large primes (LPRIME). The list of small primes is a list of all prime numbers less than 1000 that is used both in integer factorization and in polynomial factorization. The list of medium primes is a list of some number of prime numbers just less than $\beta^{1/2}$, and the list of large primes is a list of some primes just less than β. Each of these lists is generated by by a call of the subprogram DPGEN from BEGINSACLIB whenever execution of a saclib main program is initiated.

DPGEN (digit prime generator) has two inputs, m and k. It produces an ordered list of all prime numbers p such that $m \leq p < m + 2k$. DPGEN uses malloc to get an array A of k words corresponding to the k odd numbers between m and $m + 2k$ which might be primes. It initializes all array elements to 0. It then proceeds to mark (set to 1) those array elements that correspond to proper multiples of 3, since none of these are primes. Note that once the array position corresponding to a proper multiple of 3 is determined, then it and every third following array element has to be marked. Next array elements corresponding to multiples of 5, 7, 11, 13, 17, 19, 23, 25, $\ldots$ up to the square root of $m + 2k$. Note that some array elements will be marked several times, and that some of the elements in this series, for example 25, are not prime. But all nonprimes will be marked, and afterward (in Step 3), the list of primes is generated by a scan of the array.

Let us analyze the computing time of this algorithm. Marking the array for multiples of d takes time bounded by $c(k/d + 1)$ for some $c > 0$. So the total time for marking is dominated by $kH_M + M^{1/2}$, where $M = m + 2k$.