

Re: Vernon's Prime Sieve

Source: <http://sci.tech-archive.net/Archive/sci.math/2004-07/7525.html>

From: vernonner3voltazim (vnemitz_at_pinn.net)

Date: 07/29/04

Date: 28 Jul 2004 22:20:58 -0700

"Daniel McLaury" <daniel_mcl@hotmail.com> wrote:

> *I'll put in a breakpoint later today, after I watch the botball*
> *nationals.*

OK, Thanks!

> *I did a quick memory optimization of the sieve program,*
> *getting this:*

Yes, one of the things I was thinking about was reusing the buffer holding candidates for sieving. I've inserted a couple comments below

```
> /** Calculate all primes up to M * N
> *** Silly preconditions:
> *** pi(N) < P
> *** M < N
> *** SQRTN >= sqrt(N)
> *** These could be done in code, of course
> *** Total memory usage (N + P) * sizeof(int)
> ***/
>
> #define M /* Set this to whatever you want */
> #define N 65536
> #define SQRTN 256
> #define P 6600
>
> main() {
> unsigned int buffer[N], primes[P], last_prime=0;
> unsigned int i, j, k;
```

```
//From your comment at the bottom, I STARTED to suspect
// your primes[P] array is not big enough. Sure, all
// the primes that fit in 16 bits can be used to find
// the primes that fit in 32 bits, but there only
// 203million or so such primes. At four bytes each
```

```
// -- Ah, my error! printf() is going to output more
// than 4 bytes per prime, and making a 1.5GB file
// could happen easily. OK, so your program didn't
// wrap around the 32-bit mark.
```

```
> for(i = 2; i < N; i++) buffer[i] = i;
>
> for(i= 2; i < N; i++)
> if(buffer[i] && (primes[last_prime++] = i)
> && i < SQRTN)
> for(j = i * i; j < N; j += i) buffer[j] = 0;
>
> for(i = 0; i < last_prime; i++)
> printf("%d\n", primes[i]);
```

```
// I'm pretty sure you can make most parts of
// this whole program go significantly faster
// yet by using pointer arithmetic instead of
// array access. See,
```