Notice that $log\, x = u log\, y$, by the definition of u. We use the approximation for $\Psi(x,y)$ and Fact 1:

$$\begin{aligned} log\Big(\frac{\Psi(x,y)}{x}\Big) &\approx log\Big(\frac{([u]+y)!}{[u]!y!}\Big) - u\, log\, y \\ &\approx ([u]+y)log([u]+y) - ([u]+y) - \\ &\quad - \big([u]\, log\, [u] - [u]\big) - \big(y\, log\, y - y\big) - u\, log\, y. \end{aligned}$$

We now make some further approximations. First, we replace $[u]$ by u. Next, we note that, because u is assumed to be much smaller than y, we can replace $log(u+y)$ by $log\, y$. After cancellation we obtain

$$log\Big(\frac{\Psi(x,y)}{x}\Big) \approx -u\, log\, u,$$

i.e.,

$$\frac{\Psi(x,y)}{x} \approx u^{-u}.$$

For example, this says that if $x \approx 10^{48}$ and $y \approx 10^6$ as above, then the probability that a random number between 1 and x is a product of primes $\leq y$ is about 1 out of 8^8.

We are now ready to estimate the number of bit operations required to carry out the factor base algorithm described above, where for simplicity we shall suppose that our factor base B consists of the first $h = \pi(y)$ primes, i.e., all primes $\leq y$. To make our analysis easier, we shall suppose that B does not include -1, and that we consider the least positive residue (rather than the least absolute residue) of $b_i^2\ mod\ n$.

Thus, we estimate the number of bit operations required to carry out the following steps: (1) choose random numbers b_i between 1 and n and express the least positive residue of b_i^2 modulo n as a product of primes $\leq y$ if it can be so expressed, continuing until you have $\pi(y)+1$ different b_i's for which $b_i^2\ mod\ n$ is written as such a product; (2) find a set of linearly dependent rows in the corresponding $((\pi(y)+1)\times\pi(y))$-matrix of zeros and ones to obtain a congruence of the form $b^2 \equiv c^2\ mod\ n$; (3) if $b \equiv \pm c\ mod\ n$, repeat (1) and (2) with new b_i until you obtain $b^2 \equiv c^2\ mod\ n$ with $b \not\equiv \pm c\ mod\ n$, at which point find a nontrivial factor of n by computing $g.c.d.(b+c,n)$.

Assuming that the $b_i^2\ mod\ n$ (meaning least positive residue of b_i^2 modulo n) are randomly distributed between 1 and n, by the argument above we expect that it will take approximately u^u tries before we find a b_i such that $b_i^2\ mod\ n$ is a product of primes $\leq y$, where $u = log\, n/log\, y$. We will later decide how to choose y so as to minimize the length of time. The point is that choosing y large would make u^u small, and so we would frequently encounter b_i such that $b_i^2\ mod\ n$ is a product of primes $\leq y$. However, in that case the factorization of $b_i^2\ mod\, n$ into a product involving all of those primes — which we would have to do $\pi(y)+1$ times — and