Homework 5: Problem 4

William Svoboda (wsvoboda)

Collaborators: Epi Torres-Smith, Leslie Kim

Problem 4:

(A): The algorithm is first called at step j with a_0 and b_0. During this call, if the gcd is not found the algorithm will be called again. At step $j+1$, the value of $a_1 = b_0$ and $b_1 = a_0 \mod b_0$. If the algorithm is called again, at step $j+1$ we know a_2 will have the value $b_1 = a_0 \mod b_0$ and $b_2 = a_1 \mod b_1 = b_0 \mod (a_0 \mod b_0)$.

We can first observe that the modulo operation will never *increase* the value of b. Additionally, it is given that $a \geq b > 0$. This means that either $b \leq \frac{a}{2}$ or $b > \frac{a}{2}$. If we have the case where $b \leq \frac{a}{2}$, then in the next call to the algorithm we will know that $a \mod b \leq b \leq \frac{a}{2}$. If $b > \frac{a}{2}$, then $a \mod b$ will still be less than or equal to $\frac{a}{2}$. In either case, after two calls it will be guaranteed that $b_{j+2} \leq \frac{b_j}{2}$.

(B): It is given that the binary representation of an integer x is $\log_2 x + 1$. We can view the algorithm as a series of right shifts to x, with each right shift being equal to a division by 2. From the last part, we know that $b_{j+2} \leq \frac{b_j}{2}$ so there are $\log_2 b_0$ shifts. Since this division is guaranteed every two calls, there will therefore be $2 \cdot \log_2 b_0$ shifts at most. In the worst case scenario, the gcd would be equal to 1, so right shifts would need to occur the entire length of b.