1) a) The running time of algorithm A is at least $O(n^2)$ is meaningless. Formally, $O(n^2)$ states at most, covering upper bound. If an algorithm's running time is $O(n^2)$, it means the algorithm is at most quadratic. So stating an upper bound as a lower bound by "at least" is meaningless.

b) i) $2^{n+1} = O(2^n)$ ? $2^{n+1} = 2^n \cdot 2 = O(2^n)$

$\Rightarrow 2 \cdot 2^n \leq c \cdot 2^n$ where $c = 2$, for all $n \geq 0$.

So this is true.

If a function $f$ is polynomially bounded it means there exist polynomial $g$ and $h$ such that for all $x$

$$g(x) \leqslant f(x) \leqslant h(x)$$

$[\log n]! \Rightarrow n = e^k \quad (\log n)! = (\log e^k)! = k! \quad \lim_{k \to \infty} (k!) = \infty$

There is no value that limits this value from the top so not polynomial bounded.

$[\log\log n]! \Rightarrow n = e^{e^k} \quad (\log\log n)! = k! \quad e^{(e^k)} > k!$

$\log k! < \log k^k = k \log k < k^2 < e^k = \log e^{(e^k)}$ polynomial bounded