The Theory of Computation
Bernard M.E. Moret
Addison-Wesley-Longman, 1998

Solution Manual
Chapter 8, Additional Problem 8.5

1. This is just a special case of the theorem.
2. Since L is NP-complete, there exists a polynomial time transformation, u, from *Clique* to L, such that $x \in$ *Clique* iif and only if $u(x) \in L$ and $u(x)$ runs in time $q(x)$. The problem, of course, is that u is not q-sparse; however, we can decide all but a subexponential number of instances in polynomial time using $\mathscr{A}$. Thus, we let the subexponential number of "hard" instances stand, but map all polynomially decidable instances to one of two instances. Define the new polynomial time transformation u' as follows:

$$u'(x) = \begin{cases} x_1 & \text{if } \mathscr{A}(u(x)) \text{ runs in time } p \circ q \text{ and } u(x) \in L \\ x_2 & \text{if } \mathscr{A}(u(x)) \text{ runs in time } p \circ q \text{ and } u(x) \notin L \\ u(x) & \text{otherwise} \end{cases}$$

   where $x_1 \in L$ and $x_2 \notin L$. Notice that the total number of different instances produced is subexponential; indeed, u' is $(f \circ q)$-sparse. Thus we can use Berman's theorem with $\max\{q(n), f \circ q(n)\}$ as the function f and u' as the transformation t. We conclude that the complexity of *Clique* (and thus of any NP-complete language) is polynomially related to $\max\{q(n), f \circ q(n)\}$; since q is a polynomial, then this is in turn polynomially related to $\max\{f(n), n\}$, which proves our theorem.
3. If a good approximation algorithm exists for some NP-complete problem, then the complexity of that problem (and thus of all other NP-complete problems) is subexponential, so that we have NP $\subseteq$ SubExp. Further, if the subexponential function is in fact polynomial, then P = NP.