Student who wrote the solution to this question: Zimo Li, Yu-Hsuan Chuang, Jiahong Zhai
Students who read this solution to verify its clarity and correctness: Zimo Li, Yu-Hsuan Chuang, Jiahong Zhai

1. (a) Running time of merging the first two piles is

$$m + m$$

then the running time of merging the resulting pile with the third pile is

$$2m + m$$

similarly, merging all piles into a single pile would have a worst case running time of

$$\begin{aligned}(m+m) + (2m+m) + (3m+m) + \cdots + \Big((n-1)m + m\Big) &= m\sum_{i=1}^{n-1} i + (n-1)m \\ &= \frac{n(n-1)}{2}m + (n-1)m \\ &= \frac{mn^2}{2} + \frac{mn}{2} - m\end{aligned}$$

which is $\Theta(mn^2)$.

(b) Split the piles into two sets of $n/2$ piles and merge each sets recursively, then merge the resulting two piles. If the number of piles of each set is greater than 2, split them again recursively. If there are only 2 piles left, we can merge them in $O(m)$ time. The running of this algorithm is

$$T(n) = 2 \cdot T(n/2) + c \cdot m \cdot n$$

since merging two sorted piles takes time proportional to the size of the resulting pile, and that pile would have size $m \cdot n$.
Therefore, according to the Master Theorem, the running time is $\Theta(mn \log n)$.