

Problem 4-1.

a: 3

b: 4

c: 5 Correct:7

d: In practical, m gradually increasing, but k is a constant. We don't know how to choose k . If k is small, when m becomes big and increase fast, a k -size bigger resize will be useless and frequent. If k is big, when m is small and seldom increases, $k - m$ memory will be wasted.

In theoretical, if we insert n elements, the total cost will be $O(1 + k + 2k + \dots + n) = O(n^2)$ and the amortized cost will be $O(n)$, which is bad.

Addition:the computer will play more nicely with operations based around doubling (doubling is a fast operation, allocating memory blocks of sizes that are powers of two has plenty of advantages, etc)

Problem 4-2. a: 1

b: 2 Correct:3