1. \_\_init\_\_()
   1. 24 : L = F.readlines() 🡪 O(n)
   2. 26 : for line in L[2:]: 🡪 O(m-2+n) 🡪 O(m+n)
   3. 30 : data = line.split 🡪 O(O(m+n)\*12 🡪 O(m+n)
   4. 32: while data[-1] != “stock” 🡪 O(11\*O(m+n)) 🡪 O(m+n)
   5. 40, 43, 44, 45, 46: O(O(n) + O(m+n)) 🡪 O(m+n)
   6. 53 : self.\_stocks.append(stock) 🡪 O(n)
   7. Overall : O(m+n) is the worst case time complexity
2. add()
   1. 74 : self.\_stocks.append(stock) 🡪 O(n)
   2. Overall : O(n) is the worst case time complexity
3. remove()
   1. 82 : self.\_stocks.remove(Stock(“name, ignored”), aSymbol)) 🡪 O(n)
   2. Overall : O(n) is the worst case time complexity
4. getPrice()
   1. 95 : stock = self.\_findStock(aSymbol) 🡪 O(n)
      1. ( in 121 : for stock in self.\_stocks )
   2. Overall : O(n) is the worst case time complexity
5. sell()
   1. 106 : stock = self.\_findStock(aSymbol) 🡪 O(n)
      1. ( in 121 : for stock in self.\_stocks )
   2. Overall : O(n) is the worst case time complexity