一:

算法:

1. 按活动结束时间递增排序所有活动;
2. 将最后一个活动加入优化解中;
3. 从后向前遍历所有活动，选择与优化解集合兼容的、开始时间最晚的活动加入优化解集合中;
4. 遍历结束，所得优化解集合即为最大相容集合。

证明:

实际上述算法是原算法反向执行的结果，因此，原算法与上述算法等价，其总能产生最优解。

二:

1) 在当天价格高于前一天价格时进行一次交易，即前一天购入，当天卖出，否则不交易。

设 S_i 为前 i 天的最大收益

1. 记 $S_0 = 0$;

2. 遍历 $prices$ 数组，若 $prices[i] > prices[i-1]$，则 $S_i = S_{i-1} + prices[i] - prices[i-1]$，，否则 $S_i = S_{i-1}$;

3. 遍历完成，S_n 即为所求最大收益。

总计遍历一次数组，时间复杂度为 $O(n)$.

2) 在只允许两次的情况下，贪心算法无法保证收益最大化。

贪心算法并非模拟真实的交易流程。

反例数组：$prices = [7, 1, 5, 3, 6, 4, 8]$ 两次交易情况下，贪心算法结果仍为 7，而实际最大收益为 8.

三:

在每次跳跃时，计算一次跳跃落点的下一步可以到达的最远位置，贪心地选择最远位置更远的落点，依次下去直到到达最后一个点即可。