2020/03/25 Algorithm Homework

Note: When the exercise asks you to "design an algorithm for…," it always means that "designs an EFFICIENT algorithm for … and ANALYZES your algorithm". You should keep this in mind when writing solutions.

1. Give asymptotic upper and lower bounds for $T(n)$ in each of the following recurrences. Assume that $T(n)$ is constant for $n \leqq 2$. Make your bounds as tight as possible, and justify your answers.
   a. $T(n) = 2T(n/2) + n^3$
   b. $T(n) = T(9n/10) + n$
   c. $T(n) = 16T(n/4) + n^2$
   d. $T(n) = 7T(n/3) + n^2$
   e. $T(n) = 7T(n/2) + n^2$
   f. $T(n) = 2T(n/4) + \sqrt{n}$
   g. $T(n) = T(n-1) + n$
   h. $T(n) = T(\sqrt{n}) + 1$

2. **[CLRS 3rd] Problem 4-3 More recurrence examples**
   k. $\mathrm{T(n)} = \sqrt{n}T(n-1) + n$

3. **[CLRS 3rd] Exercise 4.5-4**

4. **[CLRS 3rd] Problem 2-4 Inversions**

5. **[CLRS 3rd] Exercise 4.2-3**

6. **[CLRS 3rd] Exercise 4.2-5**

7.

Professors Howard, Fine, and Howard have proposed the following "elegant" sorting algorithm:

```
STOOGE-SORT(A, i, j)
if A[i] > A[j]
   then exchange A[i] ↔ A[j]
if i + 1 ≥ j
   then return
k ← ⌊(j - i + 1)/3⌋               ▹ Round down.
STOOGE-SORT(A, i, j - k)           ▹ First two-thirds.
STOOGE-SORT(A, i + k, j)           ▹ Last two-thirds.
STOOGE-SORT(A, i, j - k)           ▹ First two-thirds again.
```

a. Argue that, if $n = length[A]$, then STOOGE-SORT(A, 1, $length[A]$) correctly sorts the input array $A[1 \quad n]$.
b. Give a recurrence for the worst-case running time of STOOGE-SORT and a tight asymptotic (Θ-notation) bound on the worst-case running time.
c. Compare the worst-case running time of STOOGE-SORT with that of insertion sort, merge sort, heapsort, and quicksort. Do the professors deserve tenure?

8. Directly solve the original stock buying problem in $\Theta(n)$ time. (要寫Code)