CSC236 – Tutorial 5: Developing Recurrences and Master Method

Exercise 1: Selection Sort

Consider the following Python code, a recursive version of selection sort:

```python
def rec_ss(A, i):
  '''
  Sorts A[i:] by recursively finding the smallest element in the remaining
  list and putting it into its correct position.
  '''
  if i < len(A) - 1:
    # Find the minimum element in A[i:]
    small = i
    for j in range(i + 1, len(A)):
      if A[j] < A[small]:
        small = j
    # Swap A[i] and A[small]
    A[i], A[small] = A[small], A[i]

    # Sort the remainder of the list
    rec_ss(A, i + 1)
```

Note that the above has an implicit base case `i == len(A) - 1`, for which it does nothing.
Analyse the asymptotic worst-case runtime of `rec_ss` in terms of the size n of the input list `A`.

Exercise 2: Master Method

For each of the following, explain why the master method **cannot** be directly used.

1. Find Θ bound for $T(n) = nT(n/2) + 2$
2. Find Θ bound for $T(n) = T(n-1) + 3n^2$
3. Find Θ bound for $T(n) = T(n/2) + \log n$
4. Find Θ bound for $T(n) = 2T(n/4) + 3T(n/3) + 1$
5. Find closed-form for $T(n) = 2T(n/2) + 4n$