Homework 2

Problem 1. Solve the following recurrences using the master method

a) \( T(n) = 2T(n/4) + 7 \).

b) \( T(n) = 3T(n/9) + \sqrt{n} \).

c) \( T(n) = 2T(n/4) + n \log n \).

d) \( T(n) = 4T(n/2) + n \).

Problem 2. Solve the following recurrence using the recursion-tree method \( T(n) = 2T(n/3) + n^2 \).

Problem 3. Let \( T(n) \) be the running time of \( \text{ALG1} \) called for \( l = 0 \) and \( r = n - 1 \). Write the recurrence for \( T(n) \), solve it and give a big-Theta bound.

```
 ALG1(A, l, r) // input: array A[l..r]
  1  if r ≤ l + 10 then return 0
  2  s = 0, n = r - l + 1
  3  for i = l to r
  4    s += A[i]
  5    s += ALG1(A, l, r - n/3)
  6    s += ALG1(A, l + n/3, r)
  7  return s
```

Problem 4. Let \( A[.\.] \) be an array of \( n \) distinct numbers. We call a pair \((A[i], A[j])\) bad, if \( i < j \) and \( A[i] > A[j] + 10 \). Design a divide-and-conquer algorithm that computes the number of bad pairs in \( A \) in \( O(n \log n) \) time.

Problem 5. Suppose that groups of 3 are used in the deterministic selection algorithm instead of groups of 5.

(a) Suppose that the algorithm recurses on the high side \( H \). Find a constant \( c \) such that \( |H| = cn + O(1) \) in the worst case. Explain why this is indeed the largest size of \( H \). Is this constant the same for the low side? Write the recursion for the worst-case running time \( T(n) \).

(b) Prove that \( T(n) = O(n \log n) \) and \( T(n) = \Omega(n \log n) \).