1. (15 marks = 2+2+2+2+2+5) Graph Theory: Exercises 34, 36, 38 in page 619. Exercises 6, 8 in page 665. Exercise 24 in page 666.

2. (30 marks) Recurrence relations: Page 471, Exercises: 4-a, 4-d, 4-g.


4. (30 marks = 10+10+10) Professor Maxell Smart designed the following algorithm:

   procedure ElegantSort (A, i, j)
   if i + 1 ≥ j then return
   k ← ⌊(j − i + 1/3)⌋
   ElegantSort(A, i, j − k) sort first 2/3 of the array
   ElegantSort(A, i + k, j) sort the last 2/3 of the array
   ElegantSort(A, i, j − k) sort the first 2/3 of the array, again

   (a) Give a recurrence relation that counts the total number of A-element comparisons (line 1 of procedure) in ElegantSort(A, 1, n).

   (b) Use the Master theorem (page 479) to provide a big-Oh estimate for the number of A-element comparisons in this algorithm for an array of length n. Note that even if $b$ is not an integer, the thesis of the Master theorem is true, which can be established by a more general proof, where each $n/b$ in the recurrence can be either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Please provide $a$, $b$, $c$ and $d$ as specified in the Master theorem, as well as the big-Oh estimate.

   (c) How does this algorithm compare with other sorting algorithms such as insertion sort, mergesort, heapsort and quicksort, in terms of the number of (A-element) comparisons used? Does Professor Smart deserve a promotion?