Homework 1

Problem 1

Find the least integer k such that $f(n)$ is $O(n^k)$ for each of the following functions. Include values for c and n_0 as described in section 3.1, page 47 of the textbook.

- $f(n) = 2n^2 + n^3 \log n$
- $f(n) = 3n^5 + (\log n)^4$
- $f(n) = (n^4 + n^2 + 1)/(n^4 + 1)$
- $f(n) = (n^3 + 5 \log n)/(n^4 + 1)$

Problem 2

You have n quarters and a balance. You know that $n-1$ quarters have the same weight, and one weighs less than the others. Give an algorithm (in pseudocode) to identify the light quarter which uses the balance only $\log_3 n$ times in the worst case.

Problem 3

Use the Master Theorem to find the asymptotic solutions for the following recurrences

- $T(n) = 7T(\frac{n}{2}) + n^2$
- $T(n) = T(\frac{n}{2}) + 1$
- $T(n) = 4T(\frac{n}{2}) + n^3$

Problem 4

$A[1...n]$ is a **sorted** array of **distinct** integers. We want to decide whether there is an index i where $A[i] = i$.

- Describe a divide-and-conquer algorithm that solves this problem.
- Use the Master Theorem to estimate the running time of the algorithm. Your algorithm should run in $O(\log n)$ time

Problem 5

Suppose you are tossing m balls into n bins. Each ball is equally likely to land in each bin, and the ball tosses are independent. What is the expected number of bins that contain exactly k balls? Use indicator random variables to find the solution.

Problem 6

Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = T(\frac{n}{2}) + n^2$. Use the substitution method to verify your answer.

Problem 7

Using Figure 1 as a model (also can be find in the textbook page 161), illustrate the operation of HEAPSORT on the array $A = [5, 13, 12, 25, 71, 37, 27, 9, 22]$.

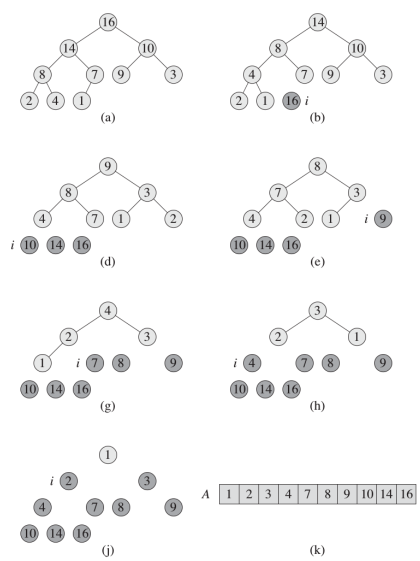

Figure 1: Example for heap sort. The original input array is [16, 14, 10, 8, 7, 9, 3, 2, 4, 1].

Problem 8

Using QUICKSORT to sort the array $A = [5, 13, 12, 25, 71, 37]$. You just need to show the result after each round. Here is a example for $A = [2, 8, 7, 1, 3, 5, 6, 4]$, suppose you pick the last element in a region as its pivot:

round 1: region=A, result=$[2, 1, 3, 4, 7, 5, 6, 8]$

round 2: region$_1$=$[2, 1, 3]$, region$_2$=$[7, 5, 6, 8]$, result=$[2, 1, 3, 4, 7, 5, 6, 8]$.

round 3: region$_1$=$[2, 1]$, region$_2$=$[7, 5, 6]$, result=$[1, 2, 3, 4, 5, 6, 7, 8]$.

The fig 2 shows the the detail operations during the round 1.

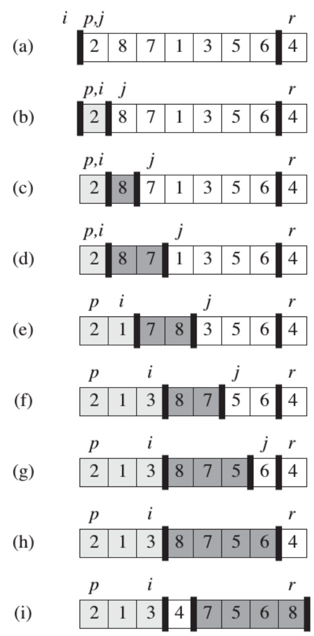

Figure 2: Example of operations in one round of quick sort. $A[r]$ is the pivot element. Elements from $A[p]$ to $A[i]$ are smaller than $A[r]$, those from $A[i + 1]$ to $A[j - 1]$ are larger than $A[r]$. In your answer you just need to show the step (i) for each round.

Problem 9

The input is two sets $S1$ and $S2$ containing n real numbers in total, and a real number x.

(a) Find a O($n$log$n$) time algorithm that determines whether there exists an element from $S1$ and an element from $S2$ whose sum is exactly x.

(b) Suppose now that the two sets are given in sorted order. Find a O(n)-time algorithm solving this problem.

You can either show pseudo code or describe it in English.

Problem 10

Show that $2n-1$ comparisons are necessary in the worst case to merge two sorted lists containing n elements each.

Problem 11 $\star$

Show an example that COUNTING SORT can be slower than any comparison sorts you have learned. (This problem is for practice, but will not be counted towards the grade)