Problem 1. Master Theorem

Here is a table of logarithms. In the row $i$ and column $j$ you find the value of $\log_i j$.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$\log_2 1 = 0$</td>
<td>$\log_2 2 = 1$</td>
<td>$\log_2 3 = 1.5849$</td>
<td>$\log_2 4 = 2$</td>
</tr>
<tr>
<td>3</td>
<td>$\log_3 1 = 0$</td>
<td>$\log_3 2 = .6309$</td>
<td>$\log_3 3 = 1$</td>
<td>$\log_3 4 = 1.2618$</td>
</tr>
<tr>
<td>4</td>
<td>$\log_4 1 = 0$</td>
<td>$\log_4 2 = .5$</td>
<td>$\log_4 3 = .7924$</td>
<td>$\log_4 4 = 1$</td>
</tr>
</tbody>
</table>

Solve the following recursions ($c$ is always a constant). Give only the final results.

(a) $T(n) = 2T(n/3) + c\sqrt{n}$.
(b) $T(n) = 4T(n/3) + cn$.
(c) $T(n) = 3T(n/3) + cn$.
(d) $T(n) = 2T(n/3) + cn$. 
Problem 2. Finding a repetition

Given a vector of integers \( X = x_1, \ldots, x_n \) we want to decide whether an element is repeated more than once.

What is the worst-case running time of the following algorithms. In the case of hash functions, specify both the average and the worst case running time. For the average case analysis, assume the existence of a universal family of functions that is computable in constant time.

(a) Assuming that the integers are in the range 1, \ldots, m, we first sort \( X \) using counting sort and then check whether two consecutive positions in the sorted vector contain the same element. (We will drop the assumption on the range in the following cases.)

(b) We use MergeSort to sort \( X \), and then proceed as before.

(c) We implement the dictionary data structure using a hash table of size \( 2^n \). The algorithm is as follows.

\[
\begin{align*}
\text{Repetitions} & \quad (X[1], \ldots, X[n]) \\
\text{begin} & \\
& \quad \text{Create empty hash table } T \text{ of size } 2n \\
& \quad \text{Insert}(X[1], T) \\
& \quad \text{for } i := 2 \text{ to } n \\
& \quad \quad \text{if } \text{Find}(X[i], T) \text{ then return true} \\
& \quad \quad \text{else } \text{Insert}(X[i], T) \\
& \quad \text{return false} \\
\text{end}
\end{align*}
\]

The algorithm creates the empty table \( T \) and does \( \text{Insert}(x_1, T) \) and then for \( i = 2, \ldots, n \) does the following: first \( \text{Find}(x_i) \), to see if \( x_i \) occurred among the previous elements; if so stops and reports a repetition. If not \( \text{Insert}(x_i, T) \) and proceeds to the next value of \( i \).
Problem 3. Median with weights

You are given a (not necessarily sorted) vector of integers \( X = x_1, \ldots, x_n \) (the integers do not come from a bounded range), and to each element \( x_i \) of the vector a weight \( w_i \) is associated (weights are non-negative integers and do not come from a bounded range). Let \( w = \sum_i w_i \).

A weighted median is a value \( v \) such that \( \sum_{i:x_i<v} w_i \leq w/2 \) and \( \sum_{i:x_i>v} w_i \leq w/2 \). Note that if \( w_i = 1 \) for all \( i \) then the weighted median is the median in the standard sense.

Show how to find the weighted median in \( O(n \log n) \) time.