Data Structures and Algorithms

Tutorial 1: Decision Trees

As preparatory work for this tutorial, it is suggested that you play a couple of games at `http://en.akinator.com/`. (Do not waste a whole day playing!) The objective of this tutorial is to get a feeling of how we might program such a game. While doing so we will manipulate binary trees: a structure that will be very important during the course.

Example set

Consider the following set of 17 vehicles and 11 questions.

	human driven?	land vehicule?	air vehicule?	water vehicule?	motorized?	powered by human/animal?	sliding vehicule?	≥ 4 wheels?	≥ 3 wheels?	≥ 2 wheels?	≥ 1 wheel?
amphibious car	1	1	0	1	1	0	0	1	1	1	1
automatic train	0	1	0	0	1	0	0	1	1	1	1
autonomous drone	0	0	1	0	1	0	0	0	1	1	1
auto-rickshaw	1	1	0	0	1	0	0	0	1	1	1
bicycle	1	1	0	0	0	1	0	0	0	1	1
bobsleigh	1	1	0	0	0	0	1	0	0	0	0
car	1	1	0	0	1	0	0	1	1	1	1
glider aircraft	1	0	1	0	0	0	0	0	1	1	1
hang glider (deltaplane)	1	0	1	0	0	0	0	0	0	0	0
helicopter	1	0	1	0	1	0	0	0	0	0	0
jumbo jet	1	0	1	0	1	0	0	1	1	1	1
monocycle	1	1	0	0	0	1	0	0	0	0	1
motorbike	1	1	0	0	1	0	0	0	0	1	1
pedalo (pedal boat)	1	0	0	1	0	1	0	0	0	1	1
rickshaw	1	1	0	0	0	1	0	0	0	1	1
sailboat	1	0	0	1	0	0	0	0	0	0	0
sleigh (horse-drawn sled)	1	1	0	0	0	1	1	0	0	0	0
wheel boat	1	0	0	1	1	0	0	0	0	1	1
yacht	1	0	0	1	1	0	0	0	0	0	0

A game

One player choses a vehicle from the set. The other player has to guess what the vehicle is by asking as few binary questions as possible (i.e. questions with yes/no answers).

Since no two vehicles have the exact set of answers for all questions, if the second player asks all 11 questions he will necessarily be able to give the answer. Your first job is to show that less than 11 questions are required.

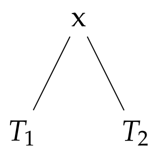
1. Draw a decision tree for this game. This is a binary tree in which each node can be labelled by
   - a question, in which case it has a left child (to follow if the answer is NO) and a right child (for YES), or
   - a answer, in which case it has no children.

   This decision tree describes an algorithm for the second player. It should start by asking the question that labels the root of the tree. And then go to one of the child according to the answer, ask the new question, and continue deeper into the decision tree until he reaches an answer.
2. Is it possible that two branches of a decision tree reach the same answer?
3. Looking at your algorithm (the decision tree you drew), calculate the number of questions you would ask in the best case scenario (less answers), worst case, and average case.
4. Assuming we always ask all questions, but not necessary in the same order, how many decision trees exist for a problem with k questions?
5. Will all decision trees have the same complexities? (By complexities we mean best, worst, and average number of questions.)
6. Can you suggest a way to order questions such that the worst case is minimized?
7. Can you suggest a way to order questions such that the best case is minimized?
8. More generally and assuming any binary question could be asked: what is the minimum possible height of a decision tree that should distinguish between n possible answers? (The height is the length of the longest branch of the three.)
9. Assuming that the decision tree is not allowed to ask questions that do not help to distinguish answers (for instance asking whether a vehicle has more that 2 wheels if the answer for ≥ 4 was NO) what is the *maximum* height of a decision tree that should distinguish n answers.

Relations on a binary tree

Binary trees are recursively defined as follows:

- The empty tree (no node), denoted λ, is a binary tree of height -1.
- Given two (disjoint) binary trees T_1 and T_2, then a node x whose left child is T_1 and whose right child is T_2 is a tree of root x, and of height $1 + \max(\mathit{height}(T_1) + \mathit{height}(T_1))$.



If y is a children of x, we say that x is y's father. The degree of a node is its number of nonempty children. A node of degree 0 is called a *leaf*, other nodes are called *internal nodes*.

1. Prove that a nonempty binary tree has n nodes of degree 2 if and only if it has $n + 1$ leaves.
2. Give the maximal number of leaves of a binary tree of height $h \geq 0$.
3. Give the minimal and maximal height of a binary tree with $f \geq 1$ leaves. (What if we disallow node of degree 1?)
4. Give the maximal number of nodes of a binary tree of height $h \geq 0$.
5. Give the minimal and maximal height of a binary tree with $n \geq 1$ nodes. (What if we disallow node of degree 1?)