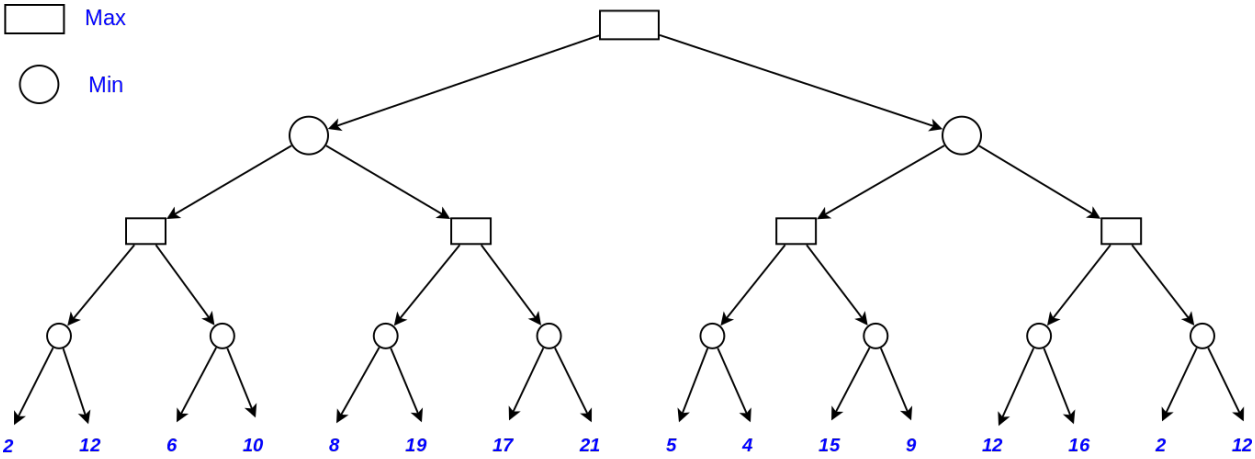
CSC520 Fall 2021 Assignment 2

Due September 20^{th} at 11:59pm

This assignment consists of three questions which involve written answers and code. In order to complete the assignment you must submit a written report in pdf form detailing your answers to the questions as well as your code. As discussed in class all work *must* be your own. You may not use third party libraries or example code to complete the assignment with the exception of csv file loaders. All reports must be clear and well written. All code must be clear, readable, and well-commented. Upload your report as a file called `<unityid>-Assign2.pdf` and your sourcecode in a zip called `<unityid>-Assign2.zip`.

1 Question 1 Adversarial Models (15 pts)



Consider the adversarial diagram above. Using the diagram answer the following questions:

1. Show the values that would be returned for each cell using min/max search.
2. Carry out $\alpha - \beta$ pruning, showing the thresholds, pruning steps with type, and returned values for each step. Number your steps to show the order of operations.
3. If you could shuffle the move order can you increase the number of items pruned? If so show the changes.

2 Question 2 Constraint Problems (10 pts)

A Golomb Ruler is a set of marks at integer positions along a number line such that no two pairs of marks are the same distance apart. The number of marks on the ruler is its order, and the largest distance between two of its marks is its length. Formulate a constraint satisfaction problem for a Golomb ruler of order o and length l. Assuming that marks can only go on integer values.

1. Identify the variables.
2. Identify the domains.
3. Give a complete specification of the constraints using mathematical and logical operators.

3 Question 3 (75 pts)

For this assignment we will be exploring bandits. You have been provided with two csv files each of which presents different sets of arms and rewards. The rewards in one file are boolean (0 or 1) while the rewards in the other range from (0 to 100). Your task is to implement three types of bandit algorithms: basic stationary, rolling average which scales the weight by the data, and the exponential recency-weighted average or the pseudostationary algorithm with a window size of 10. As your code is run it should print out: *the current step, the decision made, the current reward, and the cumulative reward.* Your code should be called as follows:

```
java -jar Bandit.jar <alg> <exp> <dist> <decay> <rwt> <w0> <infile>
```

Where:

- `alg` is one of "STAT" "ROLL" and "REC" for the three types.
- `exp` is the exploration rate. γ
- `dist` is the uniform distribution parameter, use the same distribution for all values ξ_a.
- `decay` is the decay rate β. For the nonstationary REC or nonstationary version we use this for α.
- `rwt` is the reward weight function η.
- `w0` is the initial weight value for the arms w_a^0.
- `infile` is the appropriate ad data file.

Having implemented the bandits you will need to call them on the files using several different parameter values and report how each bandit type performed under different parameter settings. Your goal in this is to assess whether the parameter values have a significant impact on the performance of the algorithms.

Note: for the purposes of this assignment you may make use of the Apache Commons CSV library in your code.

As always your code should be clear, readable, and well documented, and it should include a README file that explains how to build and execute it on the servers.