Asymptotics

Spencer Hirsch, Tyler Gutowski, Remington Greko

February 2, 2023

You have been randomly assigned to teams. Work together to write a report crossing this first bridge on algorithmic quest.

Submit the team's report on Canvas. Include a task matric indicating who did what.

Asymptotic Quest

After successful completion of these exercises you will understand the topic of *Asymptotics* and be able to explain and correctly answer questions about the topic.

The Pieces and their relationships

The pieces are functions which we will call *f, g,* and *h,* should we need others they can be named.

Standard relations include:

less than, equal, greater than, etc.

Relations can have properties such as:

Reflexing, Symmetric, Transitive

Quantifiers are also needed

For all, There exists...

Write precise (mathematical) definitions of the following relations:

1. Big-O:

   Big-O is used as a general equation in order to find the time complexity of a function. It is hardly ever exact, however, it is unnecessary for it to

be exact as a general estimate is sufficient in determining the complexity of an algorithm based on n, the number of input.

T(n) and f(n) are two positive funtcions. We can write $\mathbf{T(n) \in O(f(n))}$, and say that T(n) has order of f(n), if there are positive constants M and n_0 such that T(n) $\leq$ M * f(n) for all n $\geq n_0$

(https://yourbasic.org/algorithms/big-o-notation-explained/)

2. Big-Ω:

   Big Ω is used to give a lower bound for the gorwth of a function. It is very similar to traditional Big-O, however the inequality is different.

   T(n) and f(n) are two postivie funtions. We write $\mathbf{T(n) \in \Omega(f(n))}$, and say that T(n) is Big-Ω of f(n), if there are positve constants m and n_0 such that T(n) $\geq$ m(f(n)) for all n $\geq n_0$

3. Big-Θ:

   Now that general complexity and lower bound complexity have been defined, we can now look at Big-Θ which is used to determine both the upper and the lower bound of the time complexity function.

   $\mathbf{T(n) \in \Theta(f(n))}$ if T(n) is both OO(f(n)) and Ω(f(n))

Give examples of functions that satisfy these relations.

Explain how these relations describe bounds on running time (or other resources) expended when an algorithm is executed on input of size n.

Name	**Section**
Remington Greko	
Tyler Gutowski	
Spencer Hirsch	Mathematical definitions and explanations