CSC 236 T6: Big O

**This is a team assignment designed as an in-class activity.**

Note that this team assignment was modified from an assignment created by Clif Kussmaul, Muhlenberg College   
See <http://cspogil.org/tiki-index.php> for more information.

**Directions for use:**

* To use this form effectively, sign into a Google account.
* Then under “File” choose “Make a Copy” in order to be able to edit.
* Share with all team members, but allow Recorder to do the recording.
* Each yellow box should be filled with an appropriate team response..
* Download as *yourteamname-t6.docx* and upload to Moodle

**Big O Notation**

As you know, **Big O** is a formal way of describing an upper bound on the resources required by an algorithm for an input of a size N. An algorithm is said to **scale well** if it is suitably efficient and practical when applied to an input with a large N.

Many examples of Big O Notation can be found for algorithms which are in a mathematical domain.  But, efficient algorithms are important in all domains.  This team assignment is designed to help you to think more deeply about Big O in an other worldly domain.

In this assignment, we will look at banking procedures on some fictitious worlds.

First, confirm the roles and complete the form below for assigned roles of each member.

**Member Roles**

* If you have only four people, combine Quality Control Officer & Process Analyst
* If you have only three people, also combine Recorder & Spokesperson

|  |  |
| --- | --- |
| **Team Roles** | **Member Name** |
| **Facilitator:** | **Zach Ball** |
| **Recorder:** | **John Hellrung** |
| **Spokesperson:** | **Cody Grinnell** |
| **Quality Control Officer:** | **Angie Li** |
| **Process Analyst:** | **Angie Li** |

***Example Planet α***

Planet α uses e-currency on e-cards.  Banking acts are considered “unclean” on this planet so bankers strive for the highest levels of efficiency.  Deposits and withdrawals are both done via e-cards. If you want to withdraw 1 cubit on an e-card, it take the same time as 8 cubits or even 8 million cubits because the banker takes the same amount of time to process and swipe the e-card regardless of the amount.

Given that N is the desired withdrawal amount, compute the O(N) of the banking algorithm.

Note that in most cases it will be helpful to compute a couple of examples, but here examples are actually given in the banking description.

|  |
| --- |
| **Compute Big O Complexity (and Θ if possible)** explaining the calculation: |
| *Example*: We are told that O(1)=1, O(8)=1 and O(8,000,000)=1 ,so O(N) =1 and Θ(N) =1 no matter what N is because it takes the same time as 8 cubits or even 8 million cubits. |

Propose an appropriate name for the planet.

|  |
| --- |
| **Propose a planet name and give a reason for this name:** |
| *Example*: The planet might be named ***Constancia*** because withdrawals cost a constant amount of time. |

***Planet β***

On Planet β bankers are also clerics.  On Planet β each cubit is a heavy bag of a precious metal.  Only one of these bags can be carried at one time and they must be individually blessed by a banker before use.  The banker has to go to the vault and bring your withdrawal out and then offer an individual blessing to each bag.  Hence a withdrawal of 8 cubits takes 8 times as long as a withdrawal of 1 cubit.

Given that N is the desired withdrawal amount, compute the O(N) of the banking algorithm on Planet β after computing a couple of example withdrawals.

|  |
| --- |
| **Compute O(2) and O(3)** on this planet explaining the calculations: |
| O(2)=2  O(3)=3  The time that it takes to compute O(N) is equivalent to the value of N, the withdrawal amount. These examples make explain these calculations because they satisfy the equation O(N)=N. Each withdrawal will take 1 trip. N(1) = N. |

|  |
| --- |
| **Compute Big O Complexity (and Θ if possible)** explaining the calculation: |
| O(N)=N grows proportional to the amount withdrawn. The time it takes an individual to withdraw and receive a blessing is equivalent to the amount of units withdrawn. |

Propose an appropriate name for the planet.

|  |
| --- |
| **Propose a planet name and give a reason for this name: Linearnation** |
| Linearnation is an appropriate name because N, the desired withdrawal amount,  grows linearly in proportion with time it takes to make the withdrawals. |

***Planet γ***

On planet γ only churches may perform banking services. The fee for banking services serves as a tithe to fund projects determined to be of social benefit. For each withdrawal, a random whole number is generated which is between 0 and the withdrawal amount. The withdrawer is required to guess this number, and the withdrawal fee is based upon the number of times it takes the withdrawer to reach the correct guess. The population on the planet believes very strongly that guesses are not due to luck but that God guides each of these guesses, and the bankers only help by indicating whether each guess is correct, or high, or low.

An example transaction for a withdrawal of 2048 cubits might have a fee as low as only 1 cubit when the first guess is correct, but might have a fee as high as 10 cubits if a person was incorrect almost each time.  (For example, guesses might be 1024 (too high), 512, (too high), 256 (too high), 128 (too high), 64 (too high), 32 (too high), 16 (too high), 8 (too high), 4 (too high), 2 (too low), 3 (finally correct).

Given that N is the desired withdrawal amount, compute the O(N) of the banking algorithm on planet γ after computing a couple of example withdrawals.

|  |
| --- |
| **Compute O(8) and O(16)** on this planet explaining the calculations: |
| O(8)=3  Given the sequence “1 2 3 4 5 6 7 8” in this method taking a middle value of either 4 or 5 you could determine whether or not your estimate was too high or too low, therefore limiting the sequence to either the upper or lower half. You would continue this pattern until you eliminate 7 other options or either successfully guess the correct value. In the worst case scenario, it would take 3 guesses to select the correct value in a sequence of 8.  O(16)=4  Same process as above in a sequence of 16 that would take 4 tries in the worst case. |

|  |
| --- |
| **Compute Big O Complexity (and Θ if possible)** explaining the calculation: |
| O(N)=log base2 (N)  We halve the data with each guess. This is the inverse of multiplying by two each time (2^N). |

Propose an appropriate name for the planet.

|  |
| --- |
| **Propose a planet name and give a reason for this name: LogTutown** |
| This is the appropriate name for this planet because in order to withdraw money from they have to guess to the correct value. The worst case scenario number of guesses described as log2N. |

***Planet δ***

On planet δ all purchases must be made with a cubit of the correct size.  One may not, for example use 2 cubits marked 1 in the place of 1 cubit marked 2.

Bankers forge cubits inside a long row of numbered foundry rooms.  Each room is locked with a combination lock. Both the number of numbers in the combination as well as the highest number in the combination is the same as the denomination of cubit forged in the room.

So, for example, if you need 5 cubits, you have to go to the door which has a 5 on it and the combination lock on the door will have 5 numbers with each number being 1, 2, 3, 4, or 5. You are required to find the correct combination to retrieve a 5 cubit withdrawal, and each combination is reset after each withdrawal.

Bankers only provide access--they do not provide combinations.  The population believes that God guides the worthy in the discernment of the correct combination, so those who are less worthy may need to go through all possible combinations. For example, if you wish to withdraw 5 cubits, possible combinations include (1-2-3-4-5), (1-3-2-4-5), (1-2-4-3-5), etc.

Given that N is the desired withdrawal amount, compute the O(N) of the banking algorithm on planet δ after computing a couple of example withdrawals.

|  |
| --- |
| **Compute O(2) and O(4)** explaining the calculations: |
| O(2)=2  All possible combinations of 1 and 2 are (12) and (21). Going to the correct door is constant time so O(2)=2.  O(4)=24  The number of possible permutations of (1234) is given by 4! = 24 |

|  |
| --- |
| **Compute Big O Complexity (and Θ if possible)** explaining the calculation: |
| O(N)=N!  Factorials tell how many different ways to arrange N objects  and we are dealing with figuring out all possible combinations of (1234...N) |

Propose an appropriate name for the planet.

|  |
| --- |
| **Propose a planet name and give a reason for this name: “Exclanaccion!”** |
| Because factorials are denoted with exclamation point. |

***Planet ε***

On planet ε each cubit is made of precious metal with many unique artistic approved designs. When paying for anything if the design on each cubit is not found to be sufficiently different from the design on every other cubit, the payment will be rejected.  Hence for every withdrawal from a bank, the banker is required by law to demonstrate the significant differences of every cubit from every other.

Most bankers demonstrate using the following process: They pick a first cubit.  They compare that cubit to every other cubit in the withdrawal. When they demonstrate differences with all other cubits, they choose the second cubit from the withdrawal. They then compare the second cubit to every other cubit in the withdrawal. They continue until they have demonstrated the differences for all the cubits.

If the patron wants 8 cubits, demonstrating the first cubit takes 8 comparisons, and demonstrating the second cubit, takes 7 comparisons, etc. So, the full demonstration takes 82 or 64 comparisons.

Given that N is the desired withdrawal amount, compute the O(N) of the banking algorithm on planet ε after computing an example withdrawal.

|  |
| --- |
| **Compute O(3)** explaining the calculation: |
| O(3)=9  This calculation satisfies O(3) because with the amount input (3) we need to compared each cubit 3 times. So 3\*3=9. |

|  |
| --- |
| **Compute Big O Complexity (and Θ if possible)** explaining the calculation: |
| O(N)=N^2  We have to compare N cubits N times. N\*N=N^2 |

Propose an appropriate name for the planet.

|  |
| --- |
| **Propose a planet name and give a reason for this name: Square\_Town** |
| This is an appropriate name for this planet because its banking system requires that items requested to be withdrawn need to be compared the same number of times. Therefore the entire process is N^2. |

***Planet ζ***

On planet ζ, cubits are large and square with a hole in the middle.  They are stored in the Central Bank of Hanoi from which all withdrawals must be made in person.

Cubits are stored on a storage rod in a neat stack in ascending order of size with the smallest at the top, forming a pyramid in shape. There are two other empty storage rods in the room.

To make a withdrawal of 4 cubits you must move a stack of 4 cubits to another storage rod, obeying the following simple rules: 1) Only one cubit can be moved at a time, and cubits may only be moved to one of the other storage rods. 2) Each move consists of taking the upper cubit from one of the stacks and placing it on top of another stack i.e. a cubit can only be moved if it is the uppermost cubit on a stack. 3) No cubit may be placed on top of a smaller cubit.

Assume that N is the withdrawal amount.

|  |
| --- |
| **We will learn how to compute this Big O of this algorithm later in the course. For now, just compute at least O(2) and O(3) and describe your team’s hypotheses about the performance of the algorithm used on this planet.** |
| O(2)=3  O(3)=7  We believe that these examples represent this algorithm given that the user is trying to win. Our team experienced this by experimenting by playing the actual game with physical items and counting the turns.  (2^N) - 1 represents the number of moves to win the game(minimally I think). I learned this from a previous course. |

Propose an appropriate name for this  planet.

|  |
| --- |
| **Propose a planet name and give a reason for this name: Stackatonia 2** |
| This is an appropriate name for this planet because the algorithm that represents how people withdraw cubits is represented by (2^N) - 1. The “stack” from this planet name is appropriate because of the method that clients withdraw cubits. |

**Activity Wrap-Up**

Recall that an algorithm is said to **scale well** or be **scalable** if it is suitably efficient and practical when applied to an input with a large N.

|  |
| --- |
| **Very briefly reflect on the scalability of each of the above algorithms for large N** |
| ***Planet β:***The one with O(N)=N is does not scale well. As soon as we put in a large N, it loses efficiency since the amount of times it runs is directly proportional to the input N.  ***Planet γ:***O(N)=log base2(N)  ***Planet δ:*** O(N)=N!  ***Planet ζ:*** (2^N) - 1  ***Planet ε:*** O(N)=N^2 does not scale well. As soon as we put in a large N, it loses efficiency since the amount of times it runs is directly proportional to the input N^2. |

Complete the following table individually on the most significant or surprising thing learned in this activity by each team member.

|  |  |
| --- | --- |
| **Member Name** | **Most Significant or Most Surprising Thing Learned** |
| **Angie Li** | **I learned about the variety of situations that we could potentially encounter in algorithm design and how they can be calculated mathematically.** |
| **John Hellrung** | **Math algorithms and the underlying math behind them. The Tower of Hanoi surprising because it was the only one I understood one hundred percent and it the hardest one to wrap your head around.** |
| **Cody Grinnell** | **The Tower of Hanoi problem is also a classic problem in computer science.** |
| **Zach Ball** | **Seemingly difficult everyday problems can be broken down and described by a simple algorithm.** |

Please offer any suggestions for improvement of this activity from the team:

|  |
| --- |
| **Suggestions for improvement** |
| More representations of other algorithms. |

To submit, the Recorder will download as *yourteamname-t6.docx* and upload to Moodle while all other members will simply upload the name of the assignment (t6) and the names and roles of all team members.