Report

1. Problem Statement

By implementing the Josephus game using a list and a vector we can compare the different strengths and weaknesses of the two and compare their performances.

1. Algorithm Design

The algorithms for both implementations are very similar, with may functions being pretty much the same. In the case of eliminate next player, an iterator is set to the start of the

1. Experimental Setup

PC Specs:

Type: Laptop

**OS: Windows 10 Home**

Processor: Intel (R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHZ

**RAM: 12.0 GB (11.9 usable) (12288 MB)**

**Clock speed: (2.7GHz)**

**CPU: 4 CPUs**

System Type: 64 bit OS, X64 based processor

Condition: About 3 Yrs old (heavy use)

**Compiled and tested using PuTTY(linux) g++**

**Each experiment done once due to length of the experiments**

1. Experimental Results & Discussion

Results:

Discussion:

1. My code had a couple of unidentified bugs in the vector implementation only which seemed to only effect larger M and N tests of the vector implementation. Despite this, time dependency of N seems to affect the performance in a list more as the list must iterate through elements to access which a vector can instantly access. The M dependency affects the run time in a linear fashion. The N dependency affects the run time in a quadratic fashion (As seen in the graphs). The list seems to perform better at lower N values and is less affected by M. The vector performs better at lower M values and is less affected by N.
2. Time and elimination time vs N and M display many different trends which resemble more quadratic equations to more linear ones. This is a reflection of the fact that for a vector in the game implementation, access is linear but deletion/removal is a more quadratic time. This is the opposite for lists as removal/deletion becomes linear but access is quadratic in the game implementation.