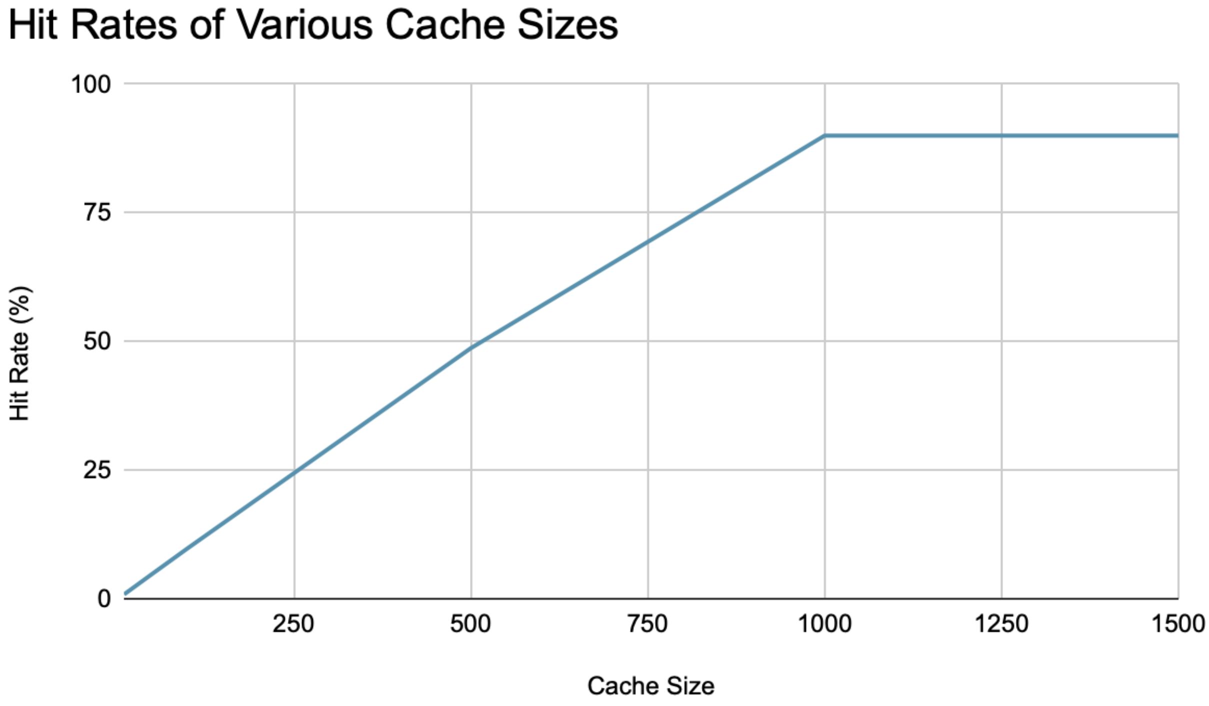
# Name: Jennah Yasin
# Date: March 28, 2022
# Title: Lab 4 - Documentation
# Description: This document will note all of the observations and
# process of implementation of Lab 4.

- In order to complete the task of this Lab, we were required to implement a successful page replacement process and ultimately show the total number of initial requests, the total number of page faults, the total number of hits, and finally the hit rate percentage based on a specific cache size. I decided to implement the task of this lab by coding in the given skeleton code. Rather than explicitly programming functions like checking if a page exists, I chose to use the given source code files (queue and node files). By creating an object in my code, I had access to various helpful functions to complete this code. I used the functions of **find(), enqueue(), dequeue()**. Once I figured this section out, all that was left were the calculations for the print statements. To figure out the total number of hits, I decided to find the difference of the total number of requests and the total number of faults, and that was fairly simple. To calculate the hit rates, I just divided the number of requests by the total hits (referred to the hit rate function). Then I just multiplied that float value by 100 to get the percentage. This lab was fairly simple to complete given that helpful resources were available to us.
- While completing the task of the Lab, we were expected to test a variation of cache sizes to understand the way cache sizes can actually affect the hit rates. I created a graph (see below) to physically compare how the hit rate is affected by different sizes of caches. I was able to ultimately understand that the larger the cache size, the higher the hit rate, which logically makes sense. That is, the more space in the cache encourages more room for hits rather than room for faults.


Hit Rates of Various Cache Sizes
Hit Rate (%)
Cache Size