검색은 정보 기술의 핵심 작업이며 우리가 매일 생성하고 처리하는 데이터의 양이 기하급수적으로 증가함에 따라 점점 더 중요해지고 있습니다. 검색 사용의 간단한 예는 맬웨어 검사와 관련이 있습니다. 각 컴퓨터 맬웨어에는 다른 맬웨어 또는 일반 코드와 구별되는 고유한 서명이 있습니다. 이 서명은 문자열, 일부 바이너리 코드 또는 멀웨어에 포함된 데이터 비트일 수 있습니다. 바이러스 스캐너는 이러한 서명을 사용하여 소프트웨어가 맬웨어에 감염되었는지 여부를 확인합니다. 현재 바이러스 서명을 생성하는 가장 간단한 방법은 MD5 또는 SHA-1과 같은 암호화 해시를 사용하는 것입니다. 해당 파일에 대한 시그니처를 사용하여 해당 파일이 알려진 맬웨어(경우에 따라 알려진 맬웨어의 변종)인지 여부를 확인하는 데 사용됩니다.

이 프로젝트의 목적은 다양한 해시/서명 저장소 유형(또는 테이블) 및 검색 알고리즘을 사용하여 성능을 구현하고 평가하는 것입니다. 8자리 숫자인 서명/해시를 사용하고 알려진 서명/해시를 정렬되지 않은 테이블(배열), 정렬된 테이블(정렬된 배열) 또는 해시 테이블에 저장하고 검색을 수행할 때 성능을 평가합니다. 이 테이블에. 따라서 수행해야 할 작업은 다음과 같습니다.

1- M개의 알려진 맬웨어 해시/시그니처를 저장하는 4개의 서로 다른 스토리지 테이블을 구현합니다. 정렬되지 않은 테이블(배열), 정렬된 테이블(정렬된 어레이) 및 2가지 유형의 해시 테이블(다음 항목 참조). 데이터 저장소는 최대 10,000개의 항목을 저장할 수 있어야 합니다.

2- 해시 테이블의 경우 해시 테이블의 요소를 추가하고 검색하기 위한 두 가지 다른 probing 접근 방식을 사용하여 두 가지 유형의 해시 테이블을 구현합니다. 하나는 linear probing을 사용하고 다른 하나는 quadratic probing을 사용합니다. 해시 테이블과 함께 모든 해시 함수를 사용하되 보고서에서 해시 함수 선택을 정당화해야 합니다.

3- M개의 해시 값을 무작위로 생성하고 각각의 사본을 다른 저장소 테이블에 저장합니다. 이 값은 알려진 맬웨어의 해시/서명을 나타냅니다.

4- 1000개의 해시 값(N)을 무작위로 생성하고 스토리지 테이블에서 이러한 값을 검색합니다.

5- 각 스토리지 유형에 대해 스토리지 생성을 위한 실행 시간을 플로팅합니다(임의로 M 해시 값을 생성하여 스토리지에 삽입). M = 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000의 다른 값에 대해. N 값을 검색하기 위한 총 실행 시간도 플로팅합니다.

6- 서로 다른 저장소에서 서로 다른 유형의 검색 알고리즘을 사용한 성능에 대한 설명합니다.

제출해야 할것들:

코드, RUN.txt라는 파일에서 코드를 실행하는 방법, 그래프를 보여주는 보고서 및 다양한 유형의 스토리지 성능에 대한 의견. 보고서에 해시 테이블과 함께 사용한 해시 함수도 표시해야 합니다.