

نحوه پیاده سازی:

پیاده سازی شامل بخش‌های زیر است:

- پیاده سازی ثوابت
- پیاده سازی تابع جستجوی خطی
- پیاده سازی تابع جستجوی دودویی
- پیاده سازی تابع جستجوی هش
- پیاده سازی تابع مقدار دهی اولیه برای بلوم سرچ
- پیاده سازی تابع هش با کمک ماژول mmh3
- پیاده سازی تابع اصلی

توضیحات پیاده سازی‌ها:

تمامی توابع جستجو، زمان فعلی را هنگام فراخوانی شدن ذخیره می‌کنند و انتها زمان کل اجرای تابع را به واحد میلی‌ثانیه همراه با یک متغیر از نوع bool بازگردانی می‌کنند.

- پیاده سازی جستجوی خطی: این تابع تک به تک عناصر را از ابتدا تا انتهای لیست چک می‌کند.
- پیاده سازی جستجوی دودویی: این تابع با استفاده از سائز دیتابیس (۴۰۳۳۵۵) هر بار بزرگی یا کوچکی عنصر هدف را با عنصر وسط مقایسه می‌کند تا در نهایت بتواند عنصر را پیدا کند.
- پیاده سازی جستجوی هش: این تابع دارای پیش نیازاتی می‌باشد
 - یک تابع هش که با کمک متغیر seed تنوع را برای هش‌های متفاوت بالا می‌برد.
 - لیست فیلتر بلوم که پیش از آن که بخواهیم جستجویی انجام دهیم باید آن ساخته شود. طریقه ساخت آن به این صورت است که در ابتدا یک لیست طبق فایل محاسبات، به طول ۴۷۸۹۶۵۶ بیت ساخته و با مقادیر ۰ پر می‌شود. سپس تک تک عناصر پایگاه داده به کمک تابع هش، ۷ بار هش شده و اندیس‌های یکسان فیلتر بلوم با خروجی تابع هش، با مقدار ۱ جایگزین می‌شود
- پس از پیاده سازی بلوم فیلتر، تنها کافی است ورودی تابع جستجوی دودویی را ۷ بار هش کنیم تا بررسی کنیم که آیا تمامی اندیس‌های یکسان فیلتر بلوم با خروجی تابع هش، برابر ۱ است یا خیر.

بررسی برتری جستجوها:

با یک مقایسه ساده بعد از اجرای برنامه خواهیم دریافت که روش جستجوی هش با فیلتر بلوم خیلی بهینه تر و سریع تر دو جستجوی دیگر مخصوصا جستجوی خطی است. زیرا مرتبه زمانی این جستجو همواره $O(7)$ است، در حالی که جستجوی دودوی $O(n \log n)$ و جستجوی خطی $O(n)$ می باشند.