

**Q1. Briefly describe the FIFO page-replacement algorithm and analyze its algorithm complexity**

使用定长队列去维护 cache,当有新 query 来时, 如果能在 cache 中找到, 则返回查询结果, hit 次数加 1; 否则, 将新 query 对应的数据入队(当队列为满时需要先出队队首元素; 否则直接入队), 本次查询为 miss。

使用 map 来维护数据是否在 cache 中, 同时记录队首队尾元素, 每一次查询与更新的时间复杂度为  $O(1)$  (认为 cache 的大小  $k$  远远小于  $n$ , 因此不考虑常数  $k$ ), 算法总时间复杂度为  $O(n)$ 。

**Q2. Briefly describe the MIN page-replacement algorithm and analyze its algorithm complexity**

当新 query 来临时, 若能在 cache 中找到, 则返回查询结果, hit 次数加 1; 否则将 query 存入 cache (当 cache 为满时 query 将替换 cache 中未来最久不会被用到的数据, 即未来最近一次查询距现在最远的数据; 否则直接存入 cache), 本次查询为 miss。MIN page-replacement 能产生最小次数的 faults。

根据离线数据获取每一个 page 下一次使用位置需要  $O(n)$  的时间。使用 map 来维护数据是否在 cache 中, 用优先队列维护 cache 中数据的下一次使用位置的最大值(即未来最久不会被用到的数据), 每一次查询与更新的复杂度为  $O(\lg n)$  (认为 cache 的大小  $k$  远远小于  $n$ , 因此不考虑常数  $k$ )。算法总时间复杂度为  $O(n \lg n)$ 。

**Q3. Briefly describe the LRU page-replacement algorithm and analyze its algorithm complexity**

当新 query 来时, 如果 cache 中能找到, 则更新相关数据结构(cache 中数据最近一次使用的记录), 返回查询结果, hit 次数加 1; 否则, 将新 query 对应的数据存入 cache (当 cache 为满时, 需要先找到 cache 中最久未用到的元素, 即最近一次使用距现在最远的数据, 用 query 替换它; 否则直接将 query 存入 cache), 更新相关数据结构, 本次查询为 miss。

使用 map 来维护数据是否在 cache 中, 使用数组维护 cache 中最久未用到的元素, 从前往后, 使用记录从远到近, 每一次取代的元素都是数组的第一个元素, 这样每一次查询与更新的复杂度为  $O(1)$  (认为 cache 的大小  $k$  远远小于  $n$ , 因此不考虑常数  $k$ ), 算法总时间复杂度为  $O(n)$ 。

**Q4. Briefly describe the clock algorithm and analyze its algorithm complexity**

使用环形列表维护 cache, 使用一个指针维护下次替换的位置。当新 query 来时, 如果 cache 中能找到, 则更新相关数据结构(将所在位置的 valid 值设置为 1), 返回查询结果, hit 次数加 1; 否则, 逐次检查指针指向的位置, 将新 query 存入 cache 中第一个遇到的 valid 为 0 的位置(若指针指向位置的 valid 值为 1, 将 valid 值置为 0, 指针向后移动一位, 直到找到一个 valid 值为 0 的位置), 将该处的 valid 重置为 1, 本次查询为 miss。

使用 map 来维护数据是否在 cache 中, 每一次查询与更新的复杂度为  $O(1)$  (认为 cache 的大小  $k$  远远小于  $n$ , 因此不考虑常数  $k$ )。算法总时间复杂度为  $O(n)$ 。

**Q5. 完成代码作业并回答 hit rate:**

1.in: FIFO: 11.98 % MIN: 42.40 % LRU: 11.76 % CLOCK: 11.93 %  
2.in: FIFO: 11.85 % MIN: 43.27 % LRU: 11.85 % CLOCK: 11.83 %  
3.in: FIFO: 82.36 % MIN: 88.58 % LRU: 82.39 % CLOCK: 82.38 %