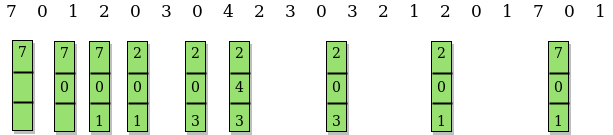
**Algorithm**

This algorithm chooses the one whose reference is farthest. But it still has excellent page fault behavior because it can communicate more memory references in less time. The optimal page replacement algorithm replaces the most recently used page with the lowest page error rate for a given page reference stream. The problem here is knowing the future perfectly.

The idea is simple, for every reference we do following :

If referred page is already present, increment hit count.

If not present, find if a page that is never referenced in future. If such a page exists, replace this page with new page. If no such page exists, find a page that is referenced farthest in future. Replace this page with new page.



The above implementation can be optimized using hashing. We can use an unordered\_set in place of vector so that search operation can be done in O(1) time.

Note that optimal page replacement algorithm is not practical as we cannot predict future. However it is used as a reference for other page replacement algorithms.