Report

150140011
O.Kürşat Karayılan

1. What is the subproblem of this greedy algorithm and what are we trying to optimize?

The problem occurs because of cache being limited and we need to evict elements from cache in a smart way when new request comes. In this example we evict furthest one in the future. We are trying to optimize choosing evicted element so that we don’t evict and take same element repeatedly because evicting and taking an element into cache is costly.

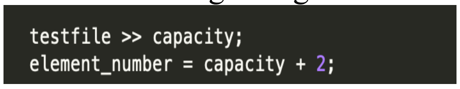
2. What is the time complexity of the given implementation in terms of n (number of requests) and m (number of elements)? Explain briefly.

We do for loop n times to check requests. When cache miss occurs, we check if the cache is full or not. To check that I checked if there are m-1(capacity) times ones or not. I do this check m times which is the size of the cache array. So, in worst case cache would be always full and we check all the time and this would make O(nm).

3. The given algorithm in the question only works for the cases where the capacity of cache is k and the number of elements is k+1.

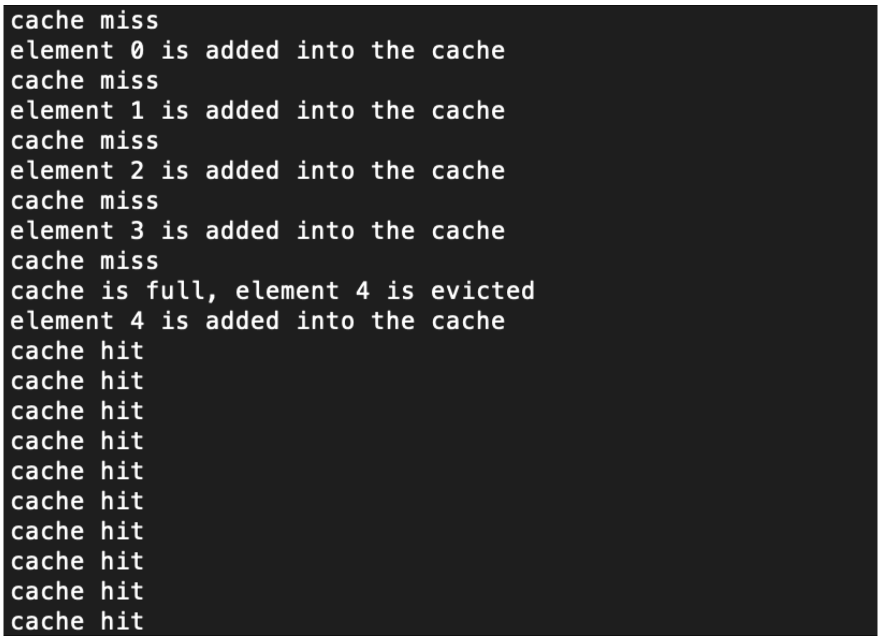
   a. Please try to show it with a counter example where the cache capacity is k and number of elements is k+2.

I made following changes.



```
testfile >> capacity;
element_number = capacity + 2;
```

And the result is:



```
cache miss
element 0 is added into the cache
cache miss
element 1 is added into the cache
cache miss
element 2 is added into the cache
cache miss
element 3 is added into the cache
cache miss
cache is full, element 4 is evicted
element 4 is added into the cache
cache hit
cache hit
cache hit
cache hit
cache hit
cache hit
cache hit
cache hit
cache hit
cache hit
```

This is not a correct result since element 4 is evicted while there is no element 4 in the cache.

b. Please briefly mention what kind of improvement would solve this problem.

When we increase element size it would create more items in linked list. But when filling fif array it would fill it with k+2th element even though we dont have it in request list. So we can delete it for once and rest would be fine.