Tanay Cansin Dogan
150150119

Analysis of Algorithms II: Q4

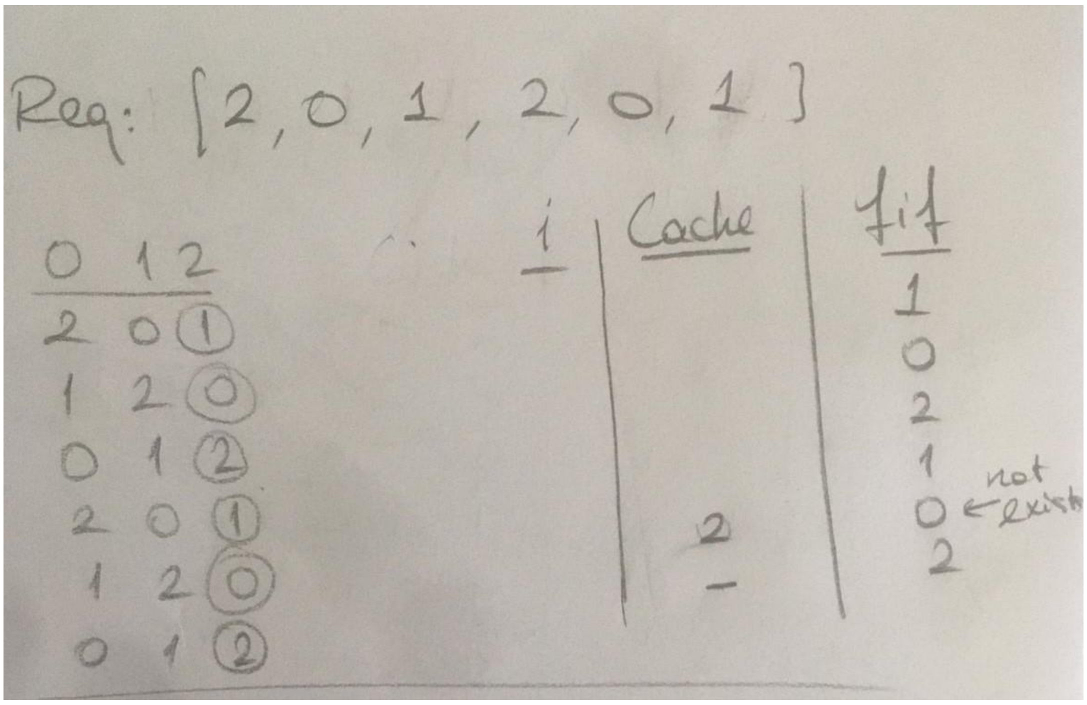
1. What is the subproblem of this greedy algorithm and what are we trying to optimize?
If we say requests: [req1:reqRest] where req1 is the first element of the requests and reqRest is the rest of the list. Subproblem of this greedy algorithm is Caching[reqRest]. And we are trying to minimize number of cache misses.

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

For the first loop: n*(removing and inserting again an element in linked list + variable assignement)
$\rightarrow$ O(n)
For the second loop: n*(if statement + search in cache + assignments)
$\rightarrow$ O(n*m)
Complexity of the algorithm = O(n*m) + O(n) = **O(n*m)**

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.



Req: [2, 0, 1, 2, 0, 1]

0	1	2
2	0	①
1	2	⓪
0	1	②
2	0	①
1	2	⓪
0	1	②

i	Cache	fif
		1
		0
		2
		1
	2	0 ← not exist
	–	2

In this counter example FiF value does not exists in cache so cannot be dismantle.

b) Please briefly mention what kind of improvement would solve this problem
using Fif[i-x] instead of Fif[i]. X where amount of deviation for example in this example x=1