Discrete Event Systems
Exercise 12

1 Bin Packing

In order to finance your studies you accept a job on the assembly line of a production firm. Your
duty is very simple: You have to pick the items delivered to you on the assembly line, put them
into a bin and close the bin.

Now assume that there are \( n \) items of size \( s_i \leq 1 \) while the bins have size 1. Moreover, assume
your algorithm is a very simple one: You handle the items in order of their arrival and put them
into a bin as long as there is enough space left. If an item arrives that does not fit into the bin
anymore, you close the bin and start with a new, empty bin.

Calculate the competitive ratio with respect to the total number of bins you need compared
to an off-line algorithm which distributes the items optimally among the bins.

2 Paging

Paging plays an important role in almost every computer system. Typically, there is a fast cache
which allows fast access, but which has limited space. On the other hand, access to the disk is
slow, but space is plenty.

We consider a simple system in which the cache has enough space to store 3 pages. Given a
request for a page \( p_i \), the system must make \( p_i \) available in the cache. If \( p_i \) is already in the cache
(called a hit), the system does nothing. Otherwise (a miss), the system incurs a page fault and
must copy the page \( p_i \) from the disk to one of the 3 locations in the cache. In case all 3 slots in
the cache are already occupied with other pages, the system is faced with the problem of which
page to evict from the cache in order to make space for \( p_i \).

In our model, we have to pay a price of 1 for each page fault, while accessing a page that is
already in the cache is for free. In this exercise, we analyze the competitiveness of several well
known paging strategies.

a) Consider the following paging strategies. Which of them are competitive and which are not?
   - FIFO (First-in/First-out): Replace the page that has been in the cache longest.
   - LFU (Least Frequently Used): Replace the page that has been requested the least since
     entering the fast memory.
   - LIFO (Last-in/First-out): Replace the page most recently moved to the cache.
   - LRU (Least Recently Used): When eviction is necessary, replace the page whose most
     recent request was the earliest.
   - FWF (Flush When Full): Whenever there is a page fault and there is no space left in
     the cache, evict all pages currently in the cache.

b) All the above strategies are deterministic. Prove a lower bound on the competitive ratio of
   any deterministic paging strategy.