Eviction policy(online)

1. 由oracle(知道未來的資訊)找出每個request的duration label(soon ,mean ,late)
2. 當某個data page進入cache後，之後又access 5\*cache size個request
   * 這裡重點是5\*cache size【個】request，強要的是【時間】，所以想表達的是【經過一段時間】
3. 位於soon, mean, late queue的data page(in cache)會被demoting(降級)，mean🡪soon, late🡪soon，代表認為在cache一段時間內的data page適合被踢掉，所以要降級
4. 但由於被降級後的data page有可能再度被存取，而LRU機制，會使被存取的data page被放在queue前端(MRU端)，因此，為了避免hot data被踢掉，在降級時，多了一項限制---如果data page位於前20%，就不降級，這是為了避免將hot data給踢掉

* 以上是online eviction policy，online 步驟如下：
  1. Request A enter RNN model
  2. RNN model預測request A 需要被cached & request A對應的duration label
  3. 但此時如果cache是滿的，就需要知道要踢掉哪個data page，在訓練時，oracle會知道每個request的benefit值，所以可以很輕易決定要踢掉哪個，可是現在是testing，我們不曉得之後request的狀況，因此求不出benefit value
  4. 所以此時會利用剛剛預測出來的duration label，依照上述方法，由soon, mean, late依序踢掉(先從soon的LRU端開始)
     + 這個方法有個好處：假設，在soon有個data page A，不斷被存取，那這個data page A就會一直處於soon queue的最前端，在select victim時，會最後被選到
     + 而由於demoting的機制，過一段時間，會將mean, late queue中，不是那麼hot(MRU前20%的data page不會做demoting)的data page往soon, mean queue丟，那挑選victim時就會優先挑到那些data page(當然前提是A一直維持在soon queue MRU端)
       - Demoting 機制還有個意涵：就是將mean, late queue中的資料往下丟，代表，那些資料好一段時間都沒有被用到(如果被用到就會跑到mean, late queue MRU端，在demoting時會特意被避開)，也就是屬於cold data，所以才要被demoting

Eviction policy(offline)