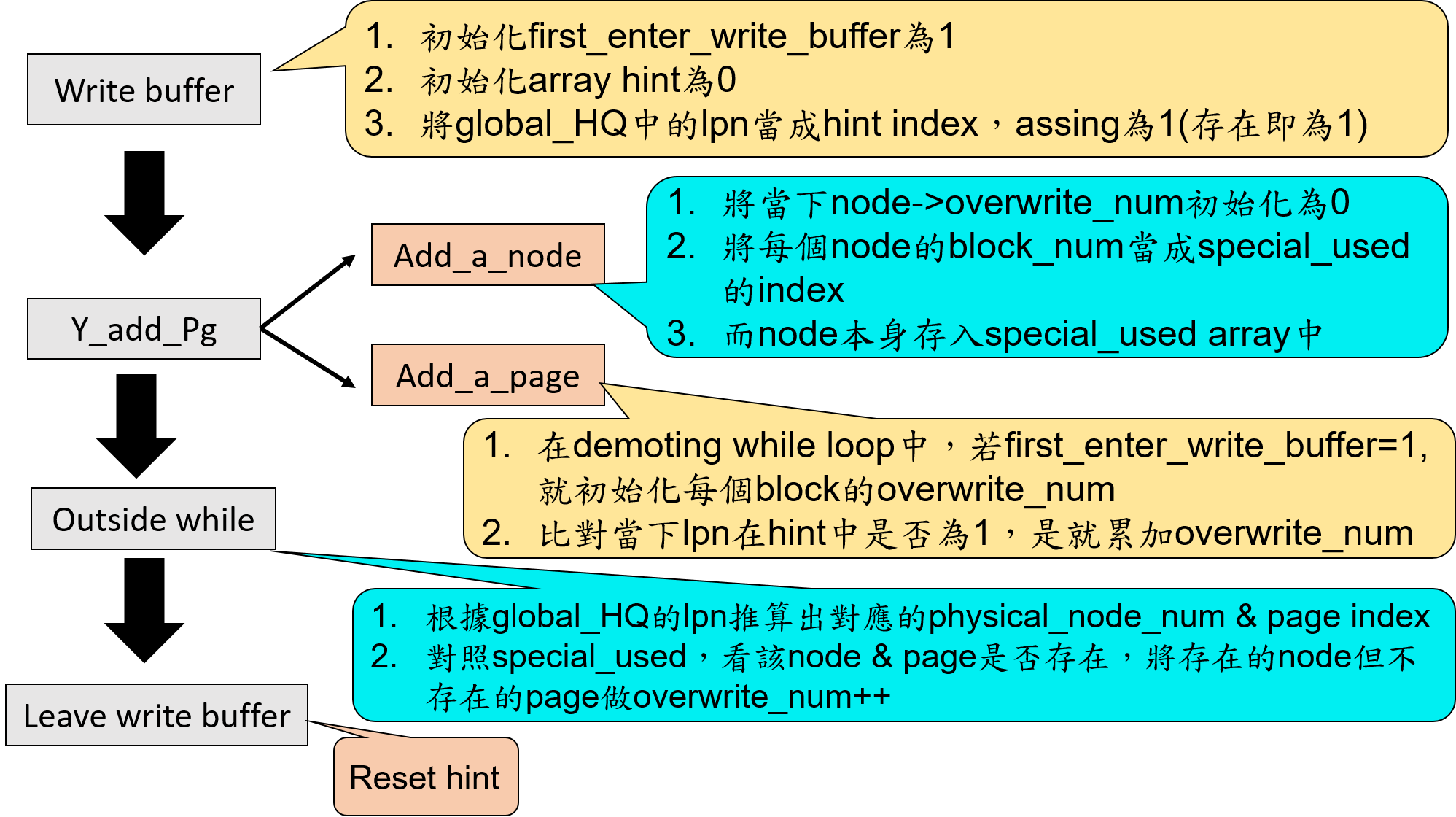
* Testing(run\_all\_trace)/AI+Hint
  + Add variable 【victim\_block】codition( if victim\_block>0 then victim\_block=0) in 【Y\_add\_Pg】 function remark part….
    1. 因為每次都會將至少一整個block remove，且remove是以block為單位，因此其實victim\_block部會被累加，但在remove mark的位置卻會被遞減，因此會出現負數。
    2. 而這個負號不斷繼續被遞減的結果，會讓後面assert(victim\_block<=64)不成立，理由還不清楚，因此在這裡設立一個條件，如果victim\_block不夠大，就不遞減。
  + Speech up victim block search time



* + 1. 在function 【add\_and\_remove】上方使用array 【hint】去記錄當下global\_HQ內所有的lpn，將lpn當成hint index，只要存在於global\_HQ，hint值就設為1；此外，同時將變數first\_enter\_write\_buffer設為1
    2. 在離開function 【add\_and\_remove】時，reset hint &變數first\_enter\_write\_buffer，因為每周期的global\_HQ內容不一樣
    3. 在一開始進入在demoting while loop內，只要first\_enter\_write\_buffer=1，就初始化所有block的overwrite\_num，完成後將first\_enter\_write\_buffer assign 0，這樣可確保overwrite\_num所計算的值屬於同一批request
    4. 在function【add\_a\_page】中，用當下lpn去檢查hint對應欄位是否為1，是，就累加該block的overwrite\_num。
    5. 在function【add\_a\_node】中，初始化overwrite\_num。
  + Add 【if original->overwrite==0 & label==soon then return】in function 【AI\_predict】
    1. 這裡主要是加快程式運作，因為如果Hint=0，基本上可以肯定當下的block就是victim block，因為會進入條件式代表比前面的benefit都要來的更小，而後面的priority一定會比較大，原因是我們是從LRU端開始掃。
    2. 所以，如果當下的Hint=0，那往MRU的方向Hint無法更小，priority又會更大的情況下，victim block就可以確定是當下的block了。
    3. 在function【Y\_add\_Pg】while loop外，根據global\_HQ的資訊抓取對應的physical\_node\_num && offset，跑過global\_HQ一遍，看special\_used中，對應的physical\_node\_num是否存在，然後對應的page[offset]是否存在
    4. 只針對存在的block中不存在的page做overwrite\_num++
* Block\_striping/AI+Hint
  + Add variable 【remove\_page\_count】and corresponding code in function 【kick\_page…】
    1. 確定至少remove 8 pages
* Testing(run\_all\_trace)/dynamic
  + In function 【kick\_page】label【Top】make sure target->logical\_node\_num<500000 if not, mark again and assign p to target.
    1. 因為target有時候會出現Invalid node，所以若發現這種狀況，就直接mark，然後assign p給target這樣就能確保數值正確。