CSE 502: Project 3 Report

LRU

- **Data Structures:**
  For the simulation of LRU, we use the java programming language. The cache is represented by a list of addresses. The address at index '0' represents the most recently used address. The address at the end of the list represents the least recently used address, and will be replaced on the next cache miss.
  We also maintain a hash set of all the addresses currently in the cache. This helps us to improve the lookup time, which is O(1) for a hash set on average. This greatly improves the running time of the simulation.
  We sore the number of hits and misses in variables of type "long".
- **Method of Simulation:**
  We first read the trace file and store all the addresses in a list in the order in which they are accessed. We then run the simulation for every address, and check if it is present in the hash set. If present, we find it in the list and move it to the MRU position in the list and increase the number of hits by 1. If it is not present, we remove the LRU element of the list if the cache is full, add the address currently accessed to the MRU position of the list, and increase the number of misses by one.
  After all the addresses have been simulated, we calculate the hit ratio using the formula:

  Hit Ratio = (Number of Hits) / (Number of Hits + Number of Misses)

ARC

- **Data Structures**
  The simulation of ARC is also done using the java programming language. We maintain four lists – t1, b1, t2 and b2. The index 0 represents the most recently used element in the list. The last element in the lists represents the least recently used element, and will be the first element to be removed from the list.
  Again, we maintain a hash set as well for all the lists. This enables fast lookup, and improves the simulation time.
  We store the number of hits and misses in variables of type "long".
- **Method of Simulation:**
  We first read the trace file and store all the addresses in a list in the order in which they are accessed. We then run the simulation for every address. If the address is present in t1 or t2, we record a hit, and move the address to the MRU position of t2. If the element is in b1 or b2, we adapt the value of p as mentioned in the algorithm, move the accessed address to the MRU position in t2 and record a miss. If the address in not present in any of the lists, we move it to the MRU position of t1 and record a miss. Every time a new element is added to t1 or t2, we remove the LRU element in one of t1 or t2, and move it to MRU position in b1 or b2 (depending on whether the element is removed from t1 or t2) and remove the LRU element from that list. For checking the correct implementation we check and make sure that the combined size of t1 and t2 never exceeds the cache size, and the combined size of t1, t2, b1 and b2 never exceeds twice the cache size.
  After all the addresses are simulated, we calculate the hit ratio using the following formula:

  Hit Ratio = (Number of Hits) / (Number of Hits + Number of Misses)