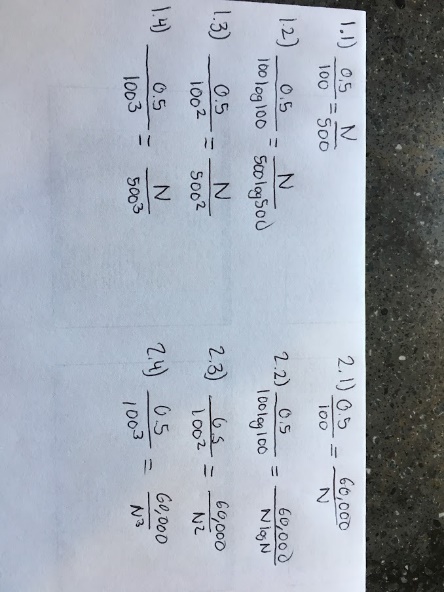
* 1. Linear: 2.5ms
  2. O(n Log n): 3.37 ms
  3. Quadratic: 12.5 ms
  4. Cubic: 62.5 ms

2.1) Linear: input size 12,000,000   
2.2) O(nLogn): input size 3,656,807  
2.3) Quadratic: input size 34,641  
2.4) Cubic: input size 4,932

3) Create empty list L3

i = 0;

j = 0;

while( i<L1.length && j < L2.length)

if(L1[i] == L2[j]

L3.add[L1[i]];

i++;

j++;

else if(L1[i]<L2[j])

i++;

else

j++;

return L3

4) Create empty list L3

i = 0;

j = 0;

while( i<L1.length || j < L2.length)

if(L1[i] == L2[j])

L3.add[L1[i]];

i++;

j++;

else if(L1[i]<L2[j])

L3.add[L1[i]];

i++;

else

L3.add[L2[i]];

j++;

return L3

5) Option B can be faster than Option A if n is less than 100 because there could be constants associated with each solution. Certain aspects of the solution are reliant on the input size while the other half could have a fixed amount of actions that need to be taken and are unaffected by input size. If Option A and Option B had the same constant added to their runtime, Option A would always be faster.