reflection

dcs229_hw4.pdf
(base) C:\Users\Drago\OneDrive\Documents\2025-classes\data_and_algs\homework4>python dcs229_hw4.py 3 1000 a 42
Using seed 42, for 3 replications of 1000 appends each,
the average loading time using append option a is 0.0000 seconds

(base) C:\Users\Drago\OneDrive\Documents\2025-classes\data_and_algs\homework4>python dcs229_hw4.py 3 1000 b 42
Using seed 42, for 3 replications of 1000 appends each,
the average loading time using append option b is 0.0000 seconds

(base) C:\Users\Drago\OneDrive\Documents\2025-classes\data_and_algs\homework4>python dcs229_hw4.py 3 1000 c 42
Using seed 42, for 3 replications of 1000 appends each,
the average loading time using append option c is 0.0000 seconds

(base) C:\Users\Drago\OneDrive\Documents\2025-classes\data_and_algs\homework4>python dcs229_hw4.py 3 10000 a 42
Using seed 42, for 3 replications of 10000 appends each,
the average loading time using append option a is 0.0052 seconds

(base) C:\Users\Drago\OneDrive\Documents\2025-classes\data_and_algs\homework4>python dcs229_hw4.py 3 10000 b 42
Using seed 42, for 3 replications of 10000 appends each,
the average loading time using append option b is 0.0052 seconds

(base) C:\Users\Drago\OneDrive\Documents\2025-classes\data_and_algs\homework4>python dcs229_hw4.py 3 10000 c 42
Using seed 42, for 3 replications of 10000 appends each,
the average loading time using append option c is 0.0781 seconds

(base) C:\Users\Drago\OneDrive\Documents\2025-classes\data_and_algs\homework4>python dcs229_hw4.py 3 100000 a 42
Using seed 42, for 3 replications of 100000 appends each,
the average loading time using append option a is 0.0573 seconds

(base) C:\Users\Drago\OneDrive\Documents\2025-classes\data_and_algs\homework4>python dcs229_hw4.py 3 100000 b 42
Using seed 42, for 3 replications of 100000 appends each,
the average loading time using append option b is 0.0625 seconds

(base) C:\Users\Drago\OneDrive\Documents\2025-classes\data_and_algs\homework4>python dcs229_hw4.py 3 100000 c 42
Using seed 42, for 3 replications of 100000 appends each,
the average loading time using append option c is 8.5573 seconds

(base) C:\Users\Drago\OneDrive\Documents\2025-classes\data_and_algs\homework4>python dcs229_hw4.py 3 1000000 a 42
Using seed 42, for 3 replications of 100000 appends each,
the average loading time using append option a is 0.0574 seconds

(base) C:\Users\Drago\OneDrive\Documents\2025-classes\data_and_algs\homework4>python dcs229_hw4.py 3 1000000 b 42
Using seed 42, for 3 replications of 100000 appends each,
the average loading time using append option b is 0.0626 seconds

(base) C:\Users\Drago\OneDrive\Documents\2025-classes\data_and_algs\homework4>python dcs229_hw4.py 3 1000000 c 42
Using seed 42, for 3 replications of 100000 appends each,
the average loading time using append option c is 20198.5573 seconds

reflection:

input size: as the number of appends increases, the time that is taken for each append (a, b, c) will increase. but the rate of increase will be different between each method because they have different implementations

a is optimized for appending to lists in python -- as input size grows, time increase linearly
b uses the extend() method -- which is efficient but slower than append because it needs to create a temporary list before appending. time increases linearly
c makes a new list each time by concatenating the list with new_list. as input size grows, the time increases quadratically making it lower for large input sizes

choice of a, b, c for given input size:
for small input sizes, the difference between methods were negligible
for larger input sizes, the difference between the methods became exponentially significant

append approach to not use:
method c is the slowest and least efficient append approach -- especially for larger input sizes. method a was slightly faster than method b but only marginally. method c is inefficient because it makes a new list every time it needs to extend the list, which means it has to copy all existing elements then add the next element (not good for larger tasks).