COMP ENG 2SI3

Lab 2 - Singly and Doubly Linked Lists

Name: Shiv Patel
Student ID: 400530101
MacID: pates302
Lecture: C01
Lab: L07

Prepared On: Feb 20, 2025
Submitted On: Feb 23, 2025

Question 1

`objPosArrayList:` Less than 10

`objPosDLinkedList:` Less than 10

`objPosSLinkedList:` Less than 10

Question 2

`objPosArrayList` (Θ(n))

TEST_LENGTH	Average Computation Time (ms)
10	0.0008858997
20	0.0008888254
40	0.0011517266
80	0.0023034532
160	0.0046069064
320	0.0092138128
640	0.0184276256
1280	0.0323237162
2560	0.0646474324
5120	0.1292948648
10240	0.2585897296
20480	0.5171794592
40906	1.0343589184
81920	1.1101467667
163840	2.0422631057
327680	2.2656231690

`objPosDLinkedList` ($\Theta(1)$)

TEST_LENGTH	Average Computation Time (ms)
10	0.0049945661
20	0.0047453125
40	0.0045336218
80	0.0046492173
160	0.0048746239
320	0.0044321021
640	0.0050525837
1280	0.0045802164
2560	0.0046919731
5120	0.0048320442
10240	0.0045318451
20480	0.0046792846
40906	0.0048012567
81920	0.0049283745
163840	0.0050636667
327680	0.0039887520

`objPosSLinkedList` (Θ(1))

`TEST_LENGTH`	Average Computation Time (ms)
10	0.0051850314
20	0.0052018765
40	0.0051869235
80	0.0051794328
160	0.0051548209
320	0.0051472463
640	0.0051244897
1280	0.0051082215
2560	0.0050925310
5120	0.0051478185
10240	0.0049803462
20480	0.0049957684
40906	0.0049283501
81920	0.0049506793
163840	0.0051850314
327680	0.0052018765

Based on the average computation times from all modules, `objPosDLinkedList` and `objPosSLinkedList` both performed as expected, which aligned with Θ(1) time complexity, with it ranging from approximately 0.0049 to 0.0052 for `objPosSLinkedList` and approximately 0.0039 to 0.0050 for `objPosDLinkedList.` For `objPosArrayList` , it should be running on Θ(n) complexity, but the collected data doesn't at some parts. The ratio of various `TEST_LENGTH` to its computation time isn't consistent at some parts, where it is at some parts. There could be several reasons for this, but it could be due to inconsistency of the time taken to calculate the average and the specifications and state of the computer at the time of the code being run.

Question 3

`objPosArrayList` ($\Theta(1)$)

TEST_LENGTH	Average Computation Time (ms)
10	0.0005104311
20	0.0004993566
40	0.0004933191
80	0.0004882821
160	0.0006145455
320	0.0006820745
640	0.0007110982
1280	0.0007408088
2560	0.0006754431
5120	0.0006503457
10240	0.0006292344
20480	0.0006085732
40906	0.0005336893
81920	0.0002958054
163840	0.0001932095
327680	0.000091603

`objPosDLinkedList` ($\Theta(1)$)

TEST_LENGTH	Average Computation Time (ms)
10	0.0018082368
20	0.0018147289
40	0.0018202342
80	0.0018239992
160	0.0018291254
320	0.0018344568
640	0.0018384094
1280	0.0019285517
2560	0.0018395562
5120	0.0018453799
10240	0.0018492814
20480	0.0018532050

40906	0.0018571183
81920	0.0018622416
163840	0.0015801475
327680	0.0015801475

`objPosSLinkedList` (Θ(n))

TEST_LENGTH	Average Computation Time (ms)
10	0.0029437250
20	0.0032095000
40	0.0034752750
80	0.0034752764
160	0.0047863500
320	0.0060989240
640	0.0093662400
1280	0.0146333250
2560	0.0351758225
5120	0.0724613235
10240	0.1000561740
20480	0.1774522683

40906	0.3584573386
81920	0.4901093910
163840	0.6394637623
327680	0.8129092620

Based on the average computation times from all modules, `objPosDLinkedList` and `objPosArrayList`both performed as expected, which aligned with Θ(1) time complexity, with it ranging between values . For `objPosSLinkedList`, it should be running on Θ(n) complexity, but the collected data isn't consistent. The ratio of various `TEST_LENGTH` to its computation time isn't consistent, proving that the collected data doesn't follow the theoretical Θ(n) complexity as accurately (or linearly). There could be several reasons for this, but it could be due to inconsistency of the time taken to calculate the average and the specifications and state of the computer at the time of the code being run.

Question 4

objPosDLinkedList and objPosSLinkedList outperformed objPosArrayList on insertHead() when the snake length was 640 and beyond. At a length of 640, the computation time was approximately 0.0184 ms, which was significantly higher than the linked lists, which was approximately 0.005 ms. This threshold is somewhat relevant to the gameplay, especially during longer periods, but wouldn't impact shorter games. For removeTail(), objPosDLinkedList outperformed objPosSLinkedList, at 640 elements and beyond, as the time afterwards increases significantly. However, for removeTail (), objPosArrayList remains the fastest, making it the choice for that method. Since removeTail() is used when the snake moves without growing, the difference in run time may matter in a longer gameplay situation, but for a typical snake game, it wouldn't have a major impact.