DSA-LAB WEEK3

## Group 4: [ Svay Monirath, ]

**Part A:**

### Q. A1

CSLL:

|  |  |
| --- | --- |
| Test No | Duration (nanoseconds) |
| 1 | 162 |
| 2 | 62 |
| 3 | 58 |
| 4 | 60 |
| 5 | 58 |
| Average: | 80 |

SLL:

|  |  |
| --- | --- |
| Test No | Duration (nanoseconds) |
| 1 | 99 |
| 2 | 53 |
| 3 | 55 |
| 4 | 64 |
| 5 | 54 |
| Average: | 63 |

Evaluation: Base on stats SLL is slightly faster than CSLL, because of the size of linked list CSLL would be faster if linked list size was bigger and bigger

### Q. A2

CSLL with prev:

|  |  |
| --- | --- |
| Test No | Duration (nanoseconds) |
| 1 | 512 |
| 2 | 165 |
| 3 | 96 |
| 4 | 88 |
| 5 | 88 |
| Average: | 189.8 |

CSLL without prev:

|  |  |
| --- | --- |
| Test No | Duration (nanoseconds) |
| 1 | 200 |
| 2 | 172 |
| 3 | 106 |
| 4 | 80 |
| 5 | 126 |
| Average: | 328.4 |

Evaluation: As expected having prev pointer makes deletion much faster since u don’t need to traverse to find the prev if you already have it in hand

### Q. A3

SLL:

|  |  |
| --- | --- |
| Test No | Duration (nanoseconds) |
| 1 | 115 |
| 2 | 71 |
| 3 | 71 |
| 4 | 71 |
| 5 | 72 |
| Average: | 80 |

CSLL:

|  |  |
| --- | --- |
| Test No | Duration (nanoseconds) |
| 1 | 94 |
| 2 | 45 |
| 3 | 43 |
| 4 | 42 |
| 5 | 42 |
| Average: | 53.2 |

Evaluation: The implementation of rotation for sll is that we basically have to make sll a temp csll then detach afterward to keep the sll style, So already have a csll makes that faster making it cost less time

### Q. A4

DLL:

|  |  |
| --- | --- |
| Test No | Duration (nanoseconds) |
| 1 | 491 |
| 2 | 320 |
| 3 | 292 |
| 4 | 98 |
| 5 | 50 |
| Average: | 250.2 |

SLL:

|  |  |
| --- | --- |
| Test No | Duration (nanoseconds) |
| 1 | 255 |
| 2 | 170 |
| 3 | 146 |
| 4 | 142 |
| 5 | 136 |
| Average: | 169.8 |

Evaluation: SLL is slightly faster than DLL at deletion since in DLL u have 1 more pointer to handle making it cost more time just reconnecting

### Q. A5

PUSH/POP front

SLL without tail:

|  |  |
| --- | --- |
| Test No | Duration (nanoseconds) |
| 1 | 195 |
| 2 | 150 |
| 3 | 167 |
| 4 | 159 |
| 5 | 136 |
| Average: | 161.4 |

SLL with tail:

|  |  |
| --- | --- |
| Test No | Duration (nanoseconds) |
| 1 | 221 |
| 2 | 163 |
| 3 | 165 |
| 4 | 162 |
| 5 | 163 |
| Average: | 174.8 |

Evaluation: The probable cost of without tail being faster than having tail is that the tail is useless if your just push/pop front the tail just cost more power on the CPU

PUSH/POP end

SLL without tail:

|  |  |
| --- | --- |
| Test No | Duration (nanoseconds) |
| 1 | 754 |
| 2 | 287 |
| 3 | 270 |
| 4 | 196 |
| 5 | 214 |
| Average: | 344.2 |

SLL with tail:

|  |  |
| --- | --- |
| Test No | Duration (nanoseconds) |
| 1 | 316 |
| 2 | 242 |
| 3 | 133 |
| 4 | 133 |
| 5 | 132 |
| Average: | 191.2 |

Evaluation: PUSH/POP operation on the end of the link-list makes having a tail much more useful and convenient making time faster

PUSH/POP front

DLL:

|  |  |
| --- | --- |
| Test No | Duration (nanoseconds) |
| 1 | 329 |
| 2 | 157 |
| 3 | 124 |
| 4 | 161 |
| 5 | 108 |
| Average: | 175.8 |

PUSH/POP end

DLL:

|  |  |
| --- | --- |
| Test No | Duration (nanoseconds) |
| 1 | 274 |
| 2 | 149 |
| 3 | 122 |
| 4 | 119 |
| 5 | 168 |
| Average: | 166.4 |

Evaluation: PUSH/POP on either front and back is relatively the same speed with DLL since you both have access to next and back

### Q. A6

N = 1000

SLL:

|  |  |
| --- | --- |
| Memory USAGE | 16000 bytes |
| Time to create | 0.98 ms |

DLL:

|  |  |
| --- | --- |
| Memory USAGE | 24000 bytes |
| Time to create | 1.39 ms |

CSLL:

|  |  |
| --- | --- |
| Memory USAGE | 16000 bytes |
| Time to create | 0.05 ms |

Evaluation: CSLL and SLL are the same size cause of the structure of node is the same while DLL has 2 pointers and why CSLL is faster could be because of the CPU warm up , but both SLL and CSLL are roughly the same

**Part B:**

### Q. B1

SLL:

|  |  |
| --- | --- |
| Test No | Duration (nanoseconds) |
| 1 | 1180 |
| 2 | 656 |
| 3 | 654 |
| 4 | 580 |
| 5 | 653 |
| Average: | 744.6 |

DLL:

|  |  |
| --- | --- |
| Test No | Duration (nanoseconds) |
| 1 | 936 |
| 2 | 618 |
| 3 | 599 |
| 4 | 598 |
| 5 | 517 |
| Average: | 653.6 |

Evaluation: SLL is slower but both are the same speed this could because of the factor that DLL has prev pointer access

### Q. B2

SLL:

|  |  |
| --- | --- |
| Test No | Duration (nanoseconds) |
| 1 | 669 |
| 2 | 462 |
| 3 | 376 |
| 4 | 375 |
| 5 | 374 |
| Average: | 451 |

Vector:

|  |  |
| --- | --- |
| Test No | Duration (nanoseconds) |
| 1 | 400 |
| 2 | 285 |
| 3 | 199 |
| 4 | 197 |
| 5 | 198 |
| Average: | 255.6 |

Evaluation: Vector is increasingly faster since all operation push pop are build in and could be because of the factor of build in function was optimize better