CSC148 Worksheet 14 Solution

Hyungmo Gu

April 24, 2020

Question 1

a.

Operation	**Running time**
Insert at the front of the list	$\mathcal{O}(n)$
Insert at the end of the list	$\mathcal{O}(1)$
Look up the element at index i, where $0 \leq i < n$	$\mathcal{O}(n)$

Correct Solution:

Operation	**Running time**
Insert at the front of the list	$\mathcal{O}(n)$
Insert at the end of the list	$\mathcal{O}(1)$
Look up the element at index i, where $0 \leq i < n$	$\mathcal{O}(1)$

b. The inserting of an element at position i requires $n - i$ elements to be shifted to right.

Using this fact, we can write the Big-Oh expression for inserting an item at index i is $\mathcal{O}(n-i)$.

Question 2

a.

Operation	**Running time**
Insert at the front of the linked list	$\mathcal{O}(1)$
Insert at the end of the linked list	$\mathcal{O}(n)$
Look up the element at index i, where $0 \leq i < n$	$\mathcal{O}(n)$

Correct Solution:

Operation	Running time
Insert at the front of the linked list	$\mathcal{O}(1)$
Insert at the end of the linked list	$\mathcal{O}(n)$
Look up the element at index i, where $0 \leq i < n$	$\mathcal{O}(i)$

b. Without the traversal, the running time of inserting is $\mathcal{O}(1)$.

With the traversal, the running time of inserting is $\mathcal{O}(i)$.

Question 3

- Unlike linked lists that store node at different memory location, array-based lists store elements in memory immediately one after another.

  Assuming it's easier for memory to find and perform operations on elements located right after another, I believe it's significantly faster for array-based lists to insert an element at position i.

Correct Solution:

Since $n - i = 1,000,000 - 500,000 = 500,000$, we can write $\mathcal{O}(n-i) \approx \mathcal{O}(i)$

Using this fact, we can conclude the speed of linked lists and array-based lists are roughly about the same.

Question 4

Question 5