CSC111 Assignment 1: Linked Lists

Zixiu Meng, Jingyang Yu

February 4, 2021

Part 1: Faster Searching in Linked Lists

1. Complete this part in the provided `a1_part1.py` starter file. Do **not** include your solution in this file.

2. Complete this part in the provided `a1_part1_test.py` starter file. Do **not** include your solution in this file.

3. (a) Let n $\in Z^+$ and $n > 2$, the list is 0,1,2...n-1
   Let m $\in Z^+$
   The for loop iterate m times
   In the iteration:

   - The first line takes 1 step
   - The while loop iterates n times, each iteration takes 1 step
   - In total, the while loop takes n steps

   Therefore, the total running time is m(n+1), which is $\Theta(mn)$ steps.

   (b) **Heuristic 1 (move to front)**
   Let n $\in Z^+$ and $n > 2$, the list is 0,1,2...n-1
   Let m $\in Z^+$
   The for loop iterates m times
   The first search: The first two line takes 1 step.
   the while loop iterate n times, each iteration has one step, then (n-1) move to the front of the list which also takes one step.
   The second search: The first two line takes 1 step.
   since (n-1) is the first, the while loop only take 1 step and the rest of code takes 1 step.In total, it is 3 steps but we can consider it as one step.
   All other search: same to the second research since the position of (n-1) does not change.
   In total, the running time will be (1+n+1)+(m-1), which is $\Theta(m + n)$
   Heuristic 2 (swap)
   case1 $m < n$
   Let n $\in Z^+$ and $n > 2$, the list is 0,1,2...n-1
   Let m $\in Z^+$
   The for loop iterates m times
   The first search: the first two lines take one step.
   The while loop iterates n times and each iteration takes 1 step.
   (n-1) swap its position with (n-2) and it takes 1 step
   The second search: the first two lines take one step
   The while loop iterates n-1 times and each iteration takes 1 step.
   (n-1) swap its position with (n-3) and it takes 1 step
   the mth search: the first two lines take one step
   the while loop iterates n-m times and each iteration takes 1 step.
   the rest code takes 1 step
   Therefore, the total running time is $\sum_{i=1}^{m}(1 + n - i + 1 + 1) = \frac{(2n-1-m)m}{2} + 3m$, which is $\Theta(mn - m^2)$, which is $\Theta(mn)$
   case2 $m >= n$ Let n $\in Z^+$, the list is 0,1,2...n-1
   Let m $\in Z^+$
   Similar to what we analyse before, in the kth ($k < n$) search, the first two lines take 1 step, the while loop

iterates (n-k+1) times, the rest code takes 1 step.
After (n-1) become the first element in the linked list, the while loop always iterates 1 step and the rest code takes constant time. Therefore, the total running time is $\sum_{i=1}^{n}(1+n-i+1+1)$+(m-n) = $\frac{(n+1)n}{2}+m+n$, which is $\Theta(n^2+m)$
Heuristic 3 (count) Let n $\in Z^+$, the list is 0,1,2...n-1 and all counts are 0
Let m $\in Z^+$
the for loop iterates m times
In the first search, the first three lines takes 1 step,
the while loop iterate n times and each iteration takes 1 step. n-1 is moved to the front of the linked list which also takes 1 step.
In the rest search, the first three lines takes 1 step,
the while loop iterate 1 times and the iteration takes 1 step. other code takes 1 step.
Therefore, the total running time is (1+n+1)+3(m-1), which is $\Theta(n+m)$

4. Let lst be a list of number [1, 2, 3, 4...n], with length of n. Let m be a sequence of (n, n-1, n-2, n-3...1, 1, 1...) with length m.
Assume $m > n$
Part1: when searching for (n, n-1, n-2...1) By comparing the code for Heuristic1 and Heuristic2, the only difference between them is the while loop because all other code takes constant time.
The first search analysis for movetofirst: the while loop iterate n times to reach n, each iteration is 1 step. Then, the linked list become (n, 1, 2...n-1).
The second search analysis movetofirst: the while loop iterate n times to reach n-1, each iteration is 1 step. Then, the linked list become (n-1, n, 1, 2...n-2).
To conclude, for Heuristic1, when searching for k, the while loop always iterates n times. Therefore, the total running time for part 1 is T1 = n^2
The first search analysis for swap: the while loop iterate n times to reach n, each iteration is 1 step. Then, the linked list becomes (1, 2,3...n, n-1)
The second search analysis for swap: the while loop iterate n times to reach n-1, each iteration is 1 step. Then, the linked list becomes (1, 2,3...n-1, n)
The third search analysis for swap: the while loop iterate n-2 times to reach n-2, each iteration is 1 step. Then, the linked list becomes (1, 2,3...n-2, n-3, n-1, n)
The fourth search analysis for swap: the while loop iterate n-2 times to reach n-3, each iteration is 1 step. Then, the linked list becomes (1, 2, 3...n-3, n-2, n-1, n)
To conclude, after each two searches, the list will become the original order. The total running time when n is even is T2 = $\sum_{i=1}^{n/2}(n-2i+2)$=n+$\frac{n^2}{2}$.
the total running time when n is odd is T2=$\frac{n^2}{2}+n-\frac{1}{2}$
Part2 when searching for (1, 1, 1, 1...) with length of (m-n), the linked list already becomes (1, 2, 3...n). If we just search the first element in the list, for both heuristic, the running time are the same.
IN conclusion T1-T2 = $\frac{n^2}{2}-n$ (n is even) or $\frac{n^2}{2}-n+\frac{1}{2}$ (n is odd), which is $\Theta(n^2)$, as needed

Part 2: Linked List Visualization

Complete this part in the provided `a1_part2.py` starter file. Do **not** include your solution in this file.