Scott Ha

CPSC 335

Assignment 1 – Algorithms

Pseudocode – LeftToRight

int main()

{

vector list O(1)

int numofdiscs, numofswaps O(1)

cout << ask for number of discs to populate the list O(1)

cin >> numofdiscs O(1)

if numofdiscs > 0 O(1)

for i < numofdiscs O(n)

list push\_back ‘d’ O(1)

list push\_back ‘l’ O(1)

else

cerr << not enough discs to populate a list correctly O(1)

display(list) O(1)

numofswaps = lefttoright(list) O(1)

display(list) O(1)

cout << display num of swaps O(1)

}

void display(list)

{

for i in list.size O(n)

cout << list[i] O(1)

}

int lefttoright(list)

{

int numofswaps = 0 O(1)

for i = 0 in list.size/2 O(n/2)

for j = i in list.size –(1+i) O(n-i)

if list[j] is ‘d’ and list[j+1] is ‘l’ O(1)

swap(list[j], list[j+1]) O(1)

numofswaps++ O(1)

return numofswaps O(1)

}

Best case performance: O(n)

Worst case performance: O(n2)

Pseudocode – Lawnmower

int main()

{

vector list O(1)

int numofdiscs, numofswaps O(1)

cout << ask for number of discs to populate the list O(1)

cin >> numofdiscs O(1)

if numofdiscs > 0 O(1)

for i < numofdiscs O(n)

list push\_back ‘d’ O(1)

list push\_back ‘l’ O(1)

else

cerr << not enough discs to populate a list correctly O(1)

display(list) O(1)

numofswaps = lawnmower(list) O(1)

display(list) O(1)

cout << display num of swaps O(1)

}

void display(list)

{

for i in list.size O(n)

cout << list[i] O(1)

}

int lawnmower(list)

{

int numofswaps = 0 O(1)

for i =0 in list.size/2 O(n/2)

if i%2 is 0 O(1)

for j = i in list.size –(1+i) O(n-i)

if list[j] is ‘d’ and list[j+1] is ‘l’ O(1)

swap(list[j], list[j+1] O(1)

numofswaps++ O(1)

else

for j = list.size-(1+i) > I O(n-i)

if list[j] is ‘d’ and list[j-1] is ‘l’ O(1)

swap(list[j], list[j-1]) O(1)

numofswaps++ O(1)

return numofswaps O(1)

}

Best case performance: O(n)

Worst case performance: O(n2)