Oberon Ilano

CS 430

Test 3

Greedy Method

1.

Function greedyMethod(amount, n)

integer S(1 : n), amount (1: n), n, length, I, j, cap

float tA

//S represents the array that holds 0 or 1//

//amount is an array that represents the amount of each program //

//length represents the length of L from the user//

//tA represent the total amount of the array amount //

cap length

for i 1 i to n increment by 1 do

S(i) 0

for i 1 i to n increment by 1 do

if amount(i) > cap

S(i) 0

else

S(i) 1

cap cap – amount(i)

tA tA + amount(i)

if i < n then

S(i) cap / amount(i)

if S(i) <> 1 then

S(i) 0

for i 1 i to n increment by 1 do

printf( S(i), amount(i))

return (tA / length) //return ratio//

end greedyMethod

2.

Test files

Nondecreasing Order:

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

Decreasing Order:

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

Duplicate:

1

31

31

19

19

22

38

9

28

25

1

25

9

28

38

Even:

2

8

10

20

4

6

12

18

14

1

Random:

1

3

6

8

10

11

12

15

16

2



3.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Author(s): Oberon Ilano, F

Assignment: Test 3, Greedy Method

Due Date: October 25, 2019

Course: CS430

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<stdio.h>

#include <stdlib.h>

void menu();

void OpenFile(int \*list, int \*size, char \*filename);

void BubbleSort(int\* list, int size);

float greedyMethod(int\* amount, int n);

int main(){

int a[101];

int count, i, j;

char choice = ' ';

float ratio;

menu();

scanf("%c", &choice);

switch(choice){

case '0': printf("Goodbye! \n"); exit (1); break;

case '1': printf("\nNon-decreasing order items\n");

OpenFile(a, &count, "Nondecreasing.txt"); break;

case '2': printf("\nDecreasing order items\n");

OpenFile(a, &count, "Decreasing.txt");break;

case '3': printf("\nDuplicate Items\n");

OpenFile(a, &count, "Duplicate.txt"); break;

case '4': printf("\nEven random items\n");

OpenFile(a, &count, "Even.txt");break;

case '5':printf("\nRandom Numbers\n");

OpenFile(a, &count, "Random.txt");break;

default:

printf("\nInvalid Choice!\n");break;

}

BubbleSort(a, count);

ratio = greedyMethod(a, count);

printf("\n\nRatio: %f", ratio);

return(0);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#Function Definition#

Name: menu

Author(s): Oberon Ilano

Purpose: Use to select option from the keyboard

Incoming: None

Outgoing: main menu option (char)

Return: None

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void menu(){

//option selection

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("Greedy Method \n");

printf("1. Nondecreasing order items\n");

printf("2. Decreasing order items\n");

printf("3. Duplicate items\n");

printf("4. Even random numbers\n");

printf("5. Random numbers\n");

printf("0. Quit Program \n");

printf("\nEnter a valid choice: ");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#Function Definition#

Name: OpenFile

Author(s): Oberon Ilano

Incoming: random numbers (int)

Outgoing: array of random numbers(int)

Return: None

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void OpenFile(int \*list, int \*size, char \*filename){

FILE \*inFile = fopen(filename, "r+");

if(inFile == NULL){

printf("File Error\n");

exit(1);

}

// read file for random numbers

\*size = 1;

while( fscanf(inFile, "%d", &list[\*size]) == 1 ||

\*size == 100){

\*size = \*size + 1;

}

fclose(inFile);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#Function Definition#

Author(s): Oberon Ilano (from CS 155 class)

Name: BubbleSort

Incoming: array of numbers(int)

Outgoing: array of numbers(int)

Return: none

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void BubbleSort(int\* list, int size){

int temp = 1, i, j;

for (i = 0; i < size; i++) {

for (j = i + 1; j < size; j++) {

if (list[i] < list[j]) {

temp = list[j];

list[j] = list[i];

list[i] = temp;

}

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#Function Definition#

Author(s): Oberon Ilano

Name: Greedy Method

Incoming: array of numbers(int)

Outgoing: array of numbers(int)

Return: ratio, float type

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

float greedyMethod(int\* amount, int n){

int S[101], length;

int i, j, cap;

float tA;

printf("\nEnter the length capacity: ");

scanf("%d", &length);

cap = length;

for (i = 1; i < n; i++)

S[i] = 0;

for (i = 1; i < n; i++) {

if (amount[i] > cap){

S[i] = 0;

}

else {

S[i] = 1;

cap = cap - amount[i];

tA = tA + amount[i];

}

}

if (i < n){

S[i] = cap / amount[i];

if (S[i] != 1)

S[i] = 0;

}

printf("\nPrograms \tAmount\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\t\*\*\*\*\*\*\*\*\n");

for (i = 1; i < n; i++)

printf("\n%d \t\t %d", S[i], amount[i]);

return tA / length;

}



4.

The time complexity for using Bubblesort for sorting the data inputs is .

Finding the largest amount of program takes

Therefore, the worst-case complexity takes