Q1

Create a array of students containing student’s position, max and min position and which student is on his left and right position. Swapping will directly find the left student and exchange their positions, their left and right students.Compare the two students’ new position with their min and max to find the new min and max.

#include <stdio.h>

#include <stdlib.h>

struct cell{

int pos;

int left;

int right;

int max;

int min;

};

void swap(int num,struct cell array[],int n){

//printf("swapping %d\n",num);

int temp=array[num].pos;//printf("pos is %d\n",temp);

int left=array[num].left;//printf("left is %d\n",left);

int right=array[num].right;//printf("right is %d\n",right);

int leftleft=array[array[num].left].left;//printf("leftleft is %d\n",leftleft);

array[num].pos=array[left].pos;

array[left].pos=temp;

//position swap done

if (leftleft!=-1) array[leftleft].right=num; //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

if (right!=n) array[right].left=left;

array[left].left=num;

array[left].right=right;

array[num].left=leftleft;

array[num].right=left;

//link done

if (array[num].pos<array[num].min) array[num].min=array[num].pos;

if (array[num].pos>array[num].max) array[num].max=array[num].pos;

if (array[left].pos<array[left].min) array[left].min=array[left].pos;

if (array[left].pos>array[left].max) array[left].max=array[left].pos;}

/\*for (int i=0;i<n;i++)

printf("after action pos %d min is %d max is %d left is %d right is %d\n",array[i].pos,array[i].min,array[i].max,array[i].left,array[i].right);

;\*/

int main(){

int n,m,swapnum;

scanf("%d%d",&n,&m);

struct cell array[n];

for (int i=0;i<n;i++)

{

array[i].pos=i;

array[i].left=i-1;

array[i].right=i+1;

array[i].max=i;

array[i].min=i;

}

//initialize done

for (int i=0;i<m;i++)

{

scanf("%d",&swapnum);

if ((swapnum>0)&&(swapnum<=n)) {if ((array[swapnum-1].pos>0)&&(array[swapnum-1].pos<n)) swap(swapnum-1,array,n);} //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

//main process done, print result now

for (int i=0;i<n;i++)

{

printf("%d %d\n",array[i].min+1,array[i].max+1);

}

return 0;

}

Q2

Use selection sort to sort an array of ‘model’. Compare the input array with the model array. If they do not match, loop to search where the number supposed in the position located. If their type is different, do direct swap. If their type is same, loop search a number with different type as a media to swap. If cannot locate a number of different type as media, it can’t be sorted.

#include <stdio.h>

#include <stdlib.h>

struct cell{

int value;

int type;

int model;

};

int total0,total1;

void swap(struct cell array[],int first,int second){

int temp=array[first].value;

int temp2=array[first].type;

array[first].value=array[second].value;

array[first].type=array[second].type;

array[second].value=temp;

array[second].type=temp2;

}

void selectionsort(struct cell array[],int length){//store thing in array[].model

total0=0;total1=0;

for (int i=0;i<length;i++){

if (array[i].type==1) total1++; else total0++;//count

for (int j=i;j<length;j++){

if (array[j].model<array[i].model){int temp=array[j].model;array[j].model=array[i].model;array[i].model=temp;}//exchange

}

}

//for (int i=0;i<length;i++){printf("%d %d %d\n",array[i].model,total0,total1);}

};

int check(struct cell array[],int length){

for (int i=0;i<length;i++)

{

if (array[i].value==array[i].model);

else if ((total0<1)||(total1<1)) return 0; //check swappable

else

{int j=i+1; while (array[j].value!=array[i].model){j++;} //find model's location

if (array[j].type!=array[i].type) {swap(array,i,j);} //opposite sign then swap

else { int k=i+1; while (array[k].type==array[i].type){k++;} //ssamae sign then find closest opposite sign for swap

swap(array,i,k);

swap (array,i,j);//next line is extra

swap(array,j,k);

}

}

// if (array[i].type==1) total1--; else total0--;

}

return 1;

};

int main(){

int totalcase,length;

scanf("%d",&totalcase);

int ansyesorno[totalcase];

for (int i=0;i<totalcase;i++){//loop for T times

scanf("%d",&length);

struct cell array[length];

for (int j=0;j<length;j++) {scanf("%d",&array[j].value);array[j].model=array[j].value;}

for (int j=0;j<length;j++) scanf("%d",&array[j].type);

selectionsort(array,length);

ansyesorno[i]=check(array,length);

}

for (int i=0;i<totalcase;i++) {if (ansyesorno[i]==1) printf("Yes\n"); else printf("No\n");}

return 0;}

Q3

Convert the character in input’s odd position into their opposite, for example ‘A’ become ‘Z’, ‘C’ become ‘X’, store them and their input position in array, then use merge sort to sort the array according to the string and output input position.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct cell{

char \*value;

int pos;

};

/\*

void selectionsort(struct cell array[],int length){

//printf("ok since now,first input is %s\n",array[0].value);

for (int i=0;i<length;i++)

{

for (int j=i;j<length;j++)

{

// printf("comparing %d and %d, length is %d\n",array[i].value,array[j].value,length);

if (strcmp(array[j].value,array[i].value)<0)

{

//printf("sorting it! doing %d and %d\n",i,j);

char \*temp=array[j].value;

int temp2=array[j].pos;

array[j].value=array[i].value;

array[j].pos=array[i].pos;

array[i].value=temp;

array[i].pos=temp2;

}

}

}

}

\*/

void Merge(struct cell \*array, struct cell \*arr1, int n1, struct cell \*arr2, int n2) {

int p1 = 0, p2 = 0, p = 0;

while (p1 < n1 && p2 < n2) {

if (strcmp(arr1[p1].value,arr2[p2].value)<=0)

array[p++] = arr1[p1++];

else

array[p++] = arr2[p2++];

}

while (p1 < n1)

array[p++] = arr1[p1++];

while (p2 < n2)

array[p++] = arr2[p2++];

}

void Divide(struct cell \*array, struct cell \*arr1, int n1, struct cell \*arr2, int n2) {

int i;

for (i = 0; i < n1; i++)

arr1[i] = array[i];

for (i = 0; i < n2; i++)

arr2[i] = array[i + n1];

}

void MergeSort(struct cell \*array,int length) {

int n1, n2;

struct cell \*arr1, \*arr2;

if (length > 1) {

n1 = length / 2;

n2 = length - n1;

arr1 = (struct cell \*)malloc(n1 \* sizeof(struct cell));

arr2 = (struct cell \*)malloc(n2 \* sizeof(struct cell));

Divide(array, arr1, n1, arr2, n2);

MergeSort(arr1, n1);

MergeSort(arr2, n2);

Merge(array, arr1, n1, arr2, n2);

free(arr1);

free(arr2);

}

}

void countvalue(struct cell array[],int length,int num){

char \*str;

str=(char \*)malloc(sizeof(char)\*length+1);

scanf("%s",str);

//printf("the string received is %s\n",str);

for (int i=0;i<length;i++){

if ((i%2)!=0) array[num].value[i]=155-str[i]; else array[num].value[i]=str[i];

}

};

int main(){

int totalinput,length;

scanf("%d%d\n",&totalinput,&length);

struct cell array[totalinput];

for (int i=0;i<totalinput;i++)

{

array[i].value=(char\*)malloc(sizeof(char)\*length+1);

countvalue(array,length,i);

array[i].pos=i;

//printf("the value calculated is %d\n",array[i].value);

}

//perform selection sort

MergeSort(array,totalinput);

//print result

for (int i=0;i<totalinput;i++)

{

printf("%d ",array[i].pos+1);

}

return 0;}