ใบงานที่ 20

เรื่อง การเรียงลำดับข้อมูล

เสนอ

อาจารย์ ปิยพล ยืนยงสถาวร

จัดทำโดย

นาย สารินทร์ อินต๊ะรักษา รหัส 65543206082-1

ใบงานนี้เป็นส่วนหนึ่งของรายวิชา โครงสร้างข้อมูลและขั้นตอนวิธี

หลักสูตรวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา

ประจำภาคที่ 1 ปีการศึกษา 2566

คำสั่ง/คำชี้แจง

- สร้างโค้ดโปรแกรมตามตัวอย่างในเอกสารประกอบการสอน
- แสดงโค้ดโปรแกรมเป็นส่วนๆ พร้อมอธิบาย
- แสดงผลการรันโปรแกรม พร้อมอธิบายการทำงาน
- สรุปผลการทดลอง

ลำดับขั้นการทดลอง

```
int Data1[MaxData], Data2[MaxData];
int N;
void PrepareRawData(int N)
{
int i;
srand(time(NULL)); //for difference random number in rand()
for (i=1;i<=N;i++)
Data1[i]=1+rand() % 99; //random difference number 1..99
}
void DispData(int Data[],int out) //Out is point of Outputted Number backward
{
int i;
for(i=1;i<=N;i++)
{
if(i<out)
printf("%2d ",Data[i]); //Show 2 width of number
else
printf("[%2d] ",Data[i]); //Show [ ] if it's Output
}
printf("\n");
}
```

- ประกาศตัวแปร Data1 และ Data2 เป็นอาร์เรย์ขนาด MaxData
- ประกาศตัวแปร N ที่จะใช้เก็บค่าขนาดของข้อมูล.
- void PrepareRawData(int N): ฟังก์ชันนี้ใช้ในการเตรียมข้อมูลแบบสุ่ม โดยรับพารามิเตอร์ N เพื่อกำหนดขนาดของข้อมูลที่ต้องการ. จากนั้นฟังก์ชันจะใช้ฟังก์ชัน srand() เพื่อกำหนดค่าเริ่มต้นของการสร้างเลขสุ่มตามเวลาปัจจุบัน
- void DispData(int Data[], int out): ฟังก์ชันนี้ใช้ในการแสดงข้อมูลที่อยู่ในอาร์เรย์ Data โดยรับพารามิเตอร์ Data เป็นอาร์เรย์ที่ต้องการแสดง และ out เป็นตำแหน่งของข้อมูลที่ต้องการให้แสดงในรูปแบบที่แตกต่าง

```
void swap(int a,int b)
{
int temp;
temp=Data2[a];
Data2[a]=Data2[b];
Data2[b]=temp;
}
int Maximum(int a, int b) //Fine Maximum from 2 Data
{
if(a>b)
return(a);
else
return(b);
}
```

นี่คือฟังก์ชันที่ใช้สลับค่าของสองตัวแปรที่มีชื่อว่า a และ b. โดยที่:

- a และ b เป็นพารามิเตอร์ของฟังก์ชันและรับค่าที่ถูกส่งเข้ามา.
- ในฟังก์ชันนี้, ค่าที่อยู่ในตัวแปร Data2 ที่มีดัชนี (index) a ถูกเก็บไว้ในตัวแปรชั่วคราว temp.
- ค่าที่อยู่ในตัวแปร Data2 ที่มีดัชนี b ถูกแทนที่ด้วยค่าจาก Data2[a].
- ค่าที่อยู่ในตัวแปร temp ถูกนำมาแทนที่ใน Data2[b].
- ฟังก์ชันนี้ทำหน้าที่สลับค่าของ Data2[a] และ Data2[b].

int Maximum(int a, int b): นี่คือฟังก์ชันที่ใช้หาค่าสูงสุด ระหว่าง a และ b. โดยที่:

- a และ b เป็นพารามิเตอร์ของฟังก์ชันและรับค่าที่ถูกส่งเข้ามา.
- ในฟังก์ชันนี้, ถ้า a มากกว่า b (ค่าที่อยู่ใน a มากกว่า), ฟังก์ชันจะส่งค่า a กลับ (return a).
- ถ้า b มากกว่าหรือเท่ากับ a, ฟังก์ชันจะส่งค่า b กลับ (return b).
- โดยฟังก์ชัน swap ใช้ในการสลับค่าของสองตัวแปรและ Maximum ใช้ในการหาค่าสูงสุดระหว่าง a และ b โดยส่งค่านี้กลับมาให้ผู้เรียกใช้ฟังก์ชัน.

```
void AdjustTree(int LastNode)
{
int i,Max,lson,rson,son;
bool result;
i=1;
result=false; // False is NOT Finish Adjustment yet
while(!result)
{
lson=(2*i); //Calculate LSon
rson=(2*i)+1; //Calculate RSon
son=0; //Set default Son
if(lson==LastNode)
{
son=1;
if(Data2[i]<Data2[lson]) //Check Father Data < LSon data ?
{
swap(i,lson);
DispData(Data2,LastNode+1); //Show each step result
}
result=true; //Finish Adjustment
}
if(rson<=LastNode)
{
son=2;
Max=Maximum(Data2[lson],Data2[rson]); //Find Maximum Data
if(Data2[i]<Max) //Check Father Data < Max ?
{
if(Max==Data2[lson]) //Max == Data Lson?
{
swap(i,lson);
DispData(Data2,LastNode+1); //Show each step result
if(rson==LastNode) //Check for Last Node
result=true; //Finish Adjustment
else
i=lson; //Let i follow to LSon
}
else //if Data RSon is Maximum
{
swap(i,rson);
DispData(Data2,LastNode+1); //Show each step
if(rson==LastNode) //Check for Last Node
result=true; //Finish Adjustment
else
i=rson; //Let i follow to RSon
}
}
else
result=true; //Finish Adjustment
}
if(son==0)
result=true; //Finish Adjustment
} //End While
printf("-------------------------------------------------Adjust Tree Finished at N=%d \n",LastNode);
} //End Fn.
```

- ประกาศตัวแปร i และ Max ในการตรวจสอบและการเปรียบเทียบข้อมูล และประกาศตัวแปร lson, rson, son, result เพื่อควบคุมการเปลี่ยนแปลงและสถานะของการปรับเรียงต้นไม้.
- กำหนด i เป็น 1 และ result เป็น false
- เริ่มลูป while จนกว่า result จะกลายเป็น true
- คำนวณดัชนีของลูกซ้ายและลูกขวาของโหนดปัจจุบัน i และกำหนด son เป็น 0
- ตรวจสอบว่าลูกซ้ายเป็นโหนดสุดท้ายหรือไม่ ถ้าใช่, จะทำการตรวจสอบค่าของโหนด i กับลูกซ้าย (Data2[i] และ Data2[lson]) และหาก Data2[i] น้อยกว่า Data2[lson] จะสลับค่า และแสดงผลลัพธ์ของข้อมูลหลังการปรับเรียงด้วย DispData.

- ตรวจสอบว่าลูกขวายังไม่เกินโหนดสุดท้าย. ถ้าเป็นเช่นนั้น, จะทำการตรวจสอบค่าของโหนด i กับลูกซ้ายและลูกขวา (Data2[i], Data2[lson], และ Data2[rson]) และหาค่าสูงสุดจากลูกซ้ายและลูกขวาด้วย Maximum.
- ตรวจสอบว่าค่าของโหนด i น้อยกว่าค่าสูงสุด Max. ถ้าเป็นเช่น, จะทำการสลับค่าของโหนด i กับค่าสูงสุด Max และแสดงผลลัพธ์ของข้อมูลหลังการปรับเรียงด้วย DispData.
- else ถ้าค่าสูงสุด Max ตรงกับลูกซ้าย Data2[lson] จะสลับค่าโหนด i กับลูกซ้าย lson และอัปเดตค่า i ให้ชี้ไปที่ลูกซ้าย lson หรือ ถ้าค่าสูงสุด Max ตรงกับลูกขวา Data2[rson] จะสลับค่าโหนด i กับลูกขวา rson และอัปเดตค่า i ให้ชี้ไปที่ลูกขวา rson.
- ตรวจสอบว่า son มีค่าเป็น 0 แสดงว่าโหนดปัจจุบันไม่มีลูก. ในกรณีนี้, การปรับเรียงถือว่าเสร็จสิ้นและ result ถูกตั้งค่าเป็น true เพื่อออกจากลูป while.
- คำสั่ง printf ใช้ในการแสดงข้อความที่บอกว่าการปรับเรียงต้นไม้เสร็จสิ้นที่ N โหนด และปิดฟังก์ชัน AdjustTree.

```
void CreateHeapTree() // Create form Data1 into Data2
{
int i,j,k,father;
bool result;
//Craete Heap Tree
Data2[1]=Data1[1]; //First node of Heap Tree
DispData(Data2,N+1); //Show each step result
for(i=2;i<=N;i++)
{
Data2[i]=Data1[i];
DispData(Data2,N+1); //Show each step result
result=true;
j=i; //set backward counter start here
while(result)
{
father=j/2; //Calculate Father
if((Data2[j]>Data2[father]) && (j>1)) //Heap tree adjusting
{
swap(j,father);
DispData(Data2,N+1); //Show each step result
j=father; //Let j follow to new Father
result=true;
}
else
result=false;
} //End While
} //End for
printf("-----------------------------------------------------Create Heap Tree Finished \n");
for(k=1;k<=N;k++) //Display Array subscript
printf("(%d) ",k);
printf("\n");
for(i=N;i>1;i--)
{
swap(1,i); //Output Root Node
DispData(Data2,i); //Show each step result
AdjustTree(i-1); //Call Adjust Heap Tree
} //End for
} //End Fn.
```

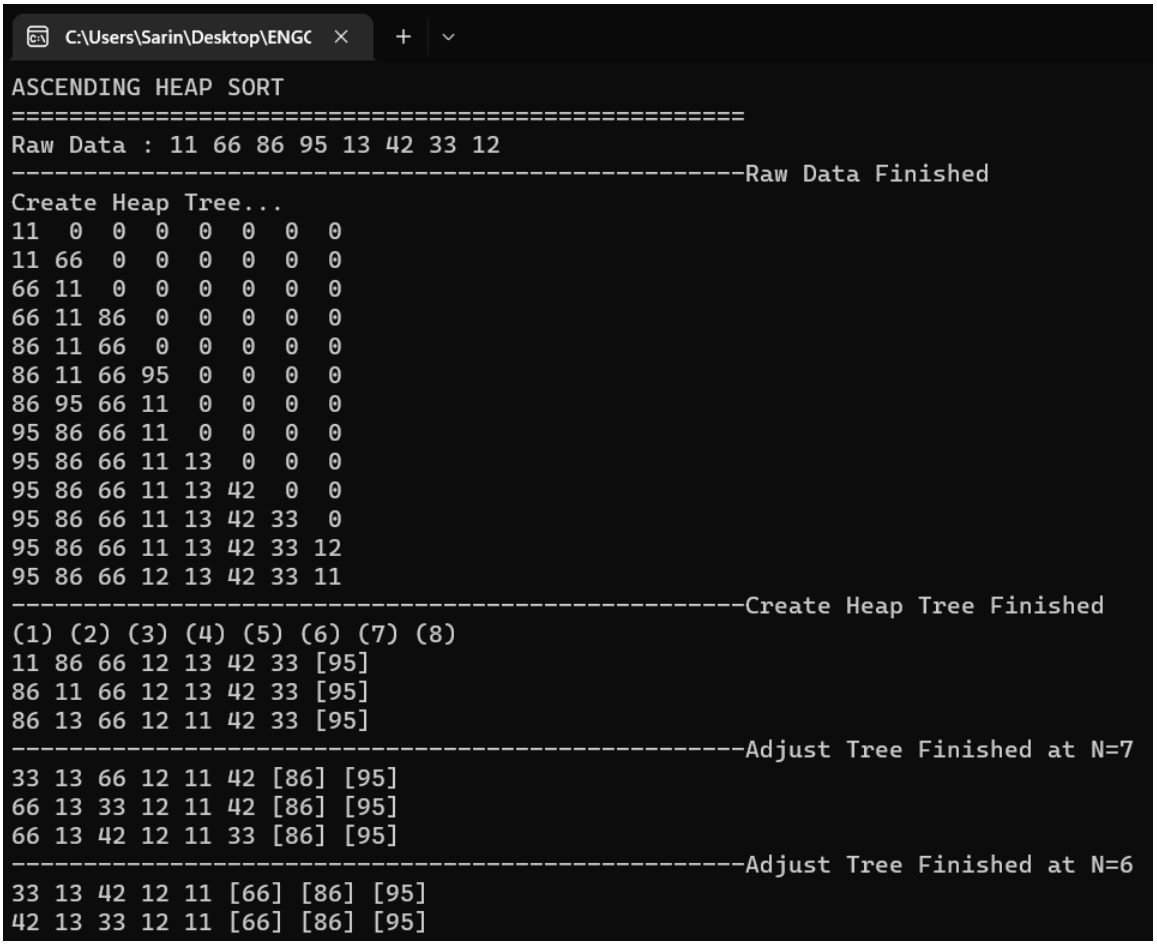
- กำหนดค่าของ Data1[1] ลงใน Data2[1] เพื่อสร้างโหนดแรกของ Heap Tree และแสดงผลลัพธ์.
- ในลูป for ที่เริ่มต้นที่ i=2 ถึง N, ฟังก์ชันทำการคัดลอกค่าจาก Data1 ไปยัง Data2 และแสดงผลลัพธ์หลังการคัดลอก. จากนั้น, มีลูป while ที่เป็นส่วนของการปรับเรียงต้นไม้ให้เป็น Heap Tree โดยสลับค่าของโหนดลูก j และโหนดพ่อ father ถ้าคุณสมบัติ Heap ไม่ถูกต้อง โดยสลับตัวแปร j ให้ชี้ไปที่โหนดพ่อใหม่ และแสดงผลลัพธ์หลังการปรับเรียง.

- ฟังก์ชันจะแสดงข้อมูลของอาร์เรย์ subscript และจะเริ่มลูป for ที่นับถอยลงมาจาก N ถึง 2. ในลูปนี้, ค่าโหนดรูท (โหนดแรก) จะถูกสลับค่ากับค่าในโหนดสุดท้ายและแสดงผลลัพธ์หลังการสลับ. จากนั้น, ฟังก์ชัน AdjustTree จะถูกเรียกใช้เพื่อปรับเรียงต้นไม้ใหม่ในลำดับที่ถูกต้องสำหรับโหนดที่เหลือที่ยังไม่ได้เรียง.

```
int main()
{
printf("ASCENDING HEAP SORT\n");
printf("==================================================\n");
N=8;
PrepareRawData(N);
printf("Raw Data : ");
DispData(Data1,N+1);
printf("--------------------------------------------------Raw Data Finished \n");
printf("Create Heap Tree...\n");
CreateHeapTree();
printf("Sorted Data is : ");
DispData(Data2,1); //Sorted Data
printf("--------------------------------------------------Sort Finished \n");
getch();
return(0);
} //End Main
```

- "ASCENDING HEAP SORT" เพื่อรายงานว่าโปรแกรมจะทำการเรียงลำดับข้อมูลแบบ Heap Sort
- กำหนดค่า N เป็น 8, ซึ่งระบุขนาดของข้อมูลที่จะถูกสร้างและเรียงลำดับ.
- เรียกใช้ฟังก์ชัน PrepareRawData(N) เพื่อสร้างข้อมูลสุ่มใน Data1
- เรียกใช้ฟังก์ชัน CreateHeapTree เพื่อสร้างต้นไม้ชนิด Heap จากข้อมูลใน Data1 และปรับเรียงให้อยู่ในลำดับที่ถูกต้อง
- แสดงข้อความ "Sorted Data is : " เพื่อบอกว่าข้อมูลที่ถูกเรียงลำดับพร้อมแสดงผลลัพธ์.
- เรียกใช้ฟังก์ชัน DispData เพื่อแสดงข้อมูลที่อยู่ใน Data2 ซึ่งเป็นข้อมูลที่เรียงลำดับแล้ว
- printf("--Sort Finished \n");:
- getch();: ใช้รอผู้ใช้กดปุ่มคีย์ใดๆ บนแป้นพิมพ์เพื่อปิดหน้าต่างโปรแกรมทันทีหลังการทำงานเสร็จสิ้น.
- return(0);: คืนค่า 0 เพื่อแสดงว่าโปรแกรมเสร็จสิ้นการทำงานโดยไม่มีข้อผิดพลาด.

ผลลัพธ์การทดลอง



สรุปผลการทดลอง

โปรแกรมนี้ใช้ในการสุ่มและเรียงลำดับข้อมูลตัวเลขใช้อัลกอริทึม HEAP SORT และแสดงผลลัพธ์ของการเรียงลำดับ

สื่อ / เอกสารอ้างอิง

ไฟล์ประกอบการสอนของ อาจารย์ ปิยพล ยืนยงสถาวร เรื่อง : การเรียงลำดับข้อมูล