



ใบงานที่ 19

เรื่อง การเรียงลำดับข้อมูล

เสนอ

อาจารย์ ปิยพล ยืนยงสถาวร

จัดทำโดย

นาย สารินทร์ อินดีะรักษา รหัส 65543206082-1

ใบงานนี้เป็นส่วนหนึ่งของรายวิชา โครงสร้างข้อมูลและขั้นตอนวิธี

หลักสูตรวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา

ประจำภาคที่ 1 ปีการศึกษา 2566

คำสั่ง/คำชี้แจง

- สร้างโค้ดโปรแกรมตามตัวอย่างในเอกสารประกอบการสอนสัปดาห์ที่7
- แสดงโค้ดโปรแกรมเป็นส่วนๆ พร้อมอธิบาย
- แสดงผลการรันโปรแกรม พร้อมอธิบายการทำงาน
- สรุปผลการทดลอง

ลำดับขั้นตอนการทดลอง

```
int Data[MaxData];
int i,N;
void PrepareRawData(int N)
{
    int i;
    srand(time(NULL)); //for difference random number in rand()
    for (i=1;i<=N;i++)
        Data[i]=1+rand() % 99; //random difference number 1..99
}
void DispData(int N)
{
    int i;
    for(i=1;i<=N;i++)
        printf(" %2d ",Data[i]);
    printf("\n");
}
void swap(int a,int b)
{
    int temp;
    temp=Data[a];
    Data[a]=Data[b];
    Data[b]=temp;
}
```

- ประกาศอาเรย์ Data เพื่อเก็บข้อมูลตัวเลข. MaxData คือขนาดสูงสุดของอาเรย์.
- ประกาศตัวแปร i และ N ซึ่งใช้ในการสร้างและจัดการข้อมูล.
- void PrepareRawData(int N) - ฟังก์ชันนี้ใช้ในการสร้างข้อมูลตัวเลขแบบสุ่มในอาเรย์ Data
- srand(time(NULL)); - ใช้เพื่อกำหนดค่าเริ่มต้นสำหรับการสุ่มตัวเลขใน rand() โดยใช้ค่าเวลาปัจจุบัน
- for (i = 1; i <= N; i++) - วนลูปเพื่อสร้างข้อมูลตัวเลข N ค่า และเก็บลงในอาเรย์ Data.
- Data[i] = 1 + rand() % 99; - สุ่มตัวเลขในช่วง 1 ถึง 99 และเก็บลงในอาเรย์ Data ในตำแหน่ง i.
- void DispData(int N) - ฟังก์ชันนี้ใช้ในการแสดงข้อมูลที่อยู่ในอาเรย์ Data โดยรับค่า N
- for (i = 1; i <= N; i++) - วนลูปเพื่อแสดงข้อมูลตัวเลข N ค่า ที่อยู่ในอาเรย์ Data.
- printf(" %2d ", Data[i]); - แสดงค่าข้อมูลในรูปแบบตัวเลขเดียว (%2d)
- printf("\n"); - แสดงขึ้นบรรทัดใหม่หลังจากแสดงข้อมูล
- void swap(int a, int b) - ฟังก์ชันนี้ใช้ในการสลับค่าข้อมูลในตำแหน่ง a และ b ของอาเรย์ Data.

```

void QuickSort(int f, int r) //Recursive Fn.
{
    int f1,r1;
    bool direction;
    f1=f; r1=r; //keep old Front & Rear values
    direction=true;
    while(f!=r)
    {
        if(Data[f]>Data[r]) //Ascending case
        { printf("%2d %2d : ",f,r);
          DispData(N);
          swap(f,r);
          printf("%2d %2d : ",f,r);
          DispData(N);
          direction= !direction; //change moving pointer direction
        }
        if (direction) //move r to Left if TRUE
            r--;
        else
            f++; //move f right if FALSE
        printf("k1=[%2d]-----\n",Data[f]); //
        //process in left hand
        if((f>f1) && (f-1 != f1))
            QuickSort(f1,f-1); //recursive new position F&R
        //process in right hand
        if((r<r1) && (r+1 != r1))
            QuickSort(r+1,r1); //recursive set new position F&R
    }
}

```

- ฟังก์ชัน Quick Sort ที่รับพารามิเตอร์ f และ r เพื่อระบุช่วงข้อมูลที่ต้องการเรียงลำดับ.
- f1 = f; r1 = r; - บันทึกค่าเริ่มต้นของตำแหน่ง f และ r
- กำหนดค่าตัวแปร direction เป็น true เพื่อระบุทิศทางการเคลื่อนที่เริ่มต้น.
- วงลูป while จนกว่าตำแหน่ง f และ r จะเข้าสู่ช่วงเดียวกัน.
- เปรียบเทียบค่าที่ตำแหน่ง f และ r และถ้าค่าที่ตำแหน่ง f มากกว่าค่าที่ตำแหน่ง r (ในรูปแบบเรียงลำดับน้อยไปมาก) จะทำการสลับค่าข้อมูลที่ตำแหน่ง f และ r.
- เรียกใช้ฟังก์ชัน swap เพื่อสลับค่าข้อมูลที่ตำแหน่ง f และ r.
- direction = !direction; - เปลี่ยนทิศทางการเคลื่อนที่ของตัวชี้ข้อมูล.
- หลังจากการสลับค่าข้อมูลและเปลี่ยนทิศทางการเคลื่อนที่ จะทำการเลื่อนตำแหน่ง f และ r ให้เข้าใกล้กันเพื่อทำการเรียงลำดับช่วงข้อมูลที่เหลือ.
- หลังจากช่วงที่เรียงลำดับเสร็จสิ้น, จะแสดงค่าที่อยู่ตำแหน่ง f.
- ตรวจสอบว่ายังมีช่วงข้อมูลที่ต้องการเรียงลำดับอยู่ในทิศทางซ้ายและทิศทางขวาของตำแหน่ง f, และถ้ามีก็จะเรียกฟังก์ชัน QuickSort ในแต่ละช่วงนี้โดยเรียกใช้เพียงครั้งเดียว

```

int main()
{
printf("ASCENDING QUICK SORT\n");
printf("=====\\n");
N=12;
PrepareRawData(N);
printf("Raw Data : ");
DispData(N);
printf("Processing Data...\\n");
printf(" F R ");
for(i=1;i<=N;i++)
printf(" (%2d)",i);
printf("\\n");
QuickSort(1,N);
printf("-----\\n");
printf("Sorted Data : ");
DispData(N); //Sorted Data
getch();
return(0);
} //End Main

```

- printf("ASCENDING QUICK SORT\\n"); - แสดงข้อความ "ASCENDING QUICK SORT"
- กำหนดค่า N เป็น 12, ซึ่งระบุจำนวนข้อมูลที่ต้องการสุมและเรียงลำดับ.
- เรียกใช้ฟังก์ชัน PrepareRawData เพื่อสุมและเตรียมข้อมูลตัวเลข.
- เรียกใช้ฟังก์ชัน DispData เพื่อแสดงข้อมูลตัวเลขที่สุมได้.
- แสดงข้อความ "Processing Data..."
- ใช้ลูป for เพื่อแสดงตำแหน่งของข้อมูลที่กำลังเรียง Quick Sort.
- เรียกใช้ฟังก์ชัน QuickSort เพื่อเรียงลำดับข้อมูลในอาร์เรย์ Data โดยให้เรียงทั้งอาร์เรย์ตั้งแต่ตำแหน่ง 1 ถึง N
- แสดงข้อความ "Sorted Data : " เพื่อบอกว่าจะแสดงข้อมูลหลังการเรียง.
- เรียกใช้ฟังก์ชัน DispData เพื่อแสดงข้อมูลหลังการเรียง Quick Sort.
- รอผู้ใช้กดปุ่มใดๆ (ใช้ getch()) เพื่อปิดหน้าต่างของโปรแกรม.
- return(0); - จบการทำงานของฟังก์ชัน main และโปรแกรม.

ผลลัพธ์การทดลอง

```
C:\Users\Sarin\Desktop\ENGC × + ✓

ASCENDING QUICK SORT
=====
=====
Raw Data :   5  46  92  24  13  37  21  66   4  45  68  68
Processing Data...
F R  ( 1) ( 2) ( 3) ( 4) ( 5) ( 6) ( 7) ( 8) ( 9) (10) (11) (12)
1  9 :   5  46  92  24  13  37  21  66   4  45  68  68
1  9 :   4  46  92  24  13  37  21  66   5  45  68  68
2  9 :   4  46  92  24  13  37  21  66   5  45  68  68
2  9 :   4   5  92  24  13  37  21  66  46  45  68  68
k1=[ 5]-----
3 12 :   4   5  92  24  13  37  21  66  46  45  68  68
3 12 :   4   5  68  24  13  37  21  66  46  45  68  92
k1=[92]-----
3 10 :   4   5  68  24  13  37  21  66  46  45  68  92
3 10 :   4   5  45  24  13  37  21  66  46  68  68  92
k1=[68]-----
3  7 :   4   5  45  24  13  37  21  66  46  68  68  92
3  7 :   4   5  21  24  13  37  45  66  46  68  68  92
k1=[45]-----
3  5 :   4   5  21  24  13  37  45  66  46  68  68  92
3  5 :   4   5  13  24  21  37  45  66  46  68  68  92
4  5 :   4   5  13  24  21  37  45  66  46  68  68  92
4  5 :   4   5  13  21  24  37  45  66  46  68  68  92
```

สรุปผลการทดลอง

โปรแกรมนี้ใช้ในการสุ่มและเรียงลำดับข้อมูลตัวเลขแบบน้อยไปหามากโดยใช้อัลกอริทึม Quick Sort และแสดงผลลัพธ์แต่ละขั้นตอนของการเรียงลำดับ.

สื่อ / เอกสารอ้างอิง

ไฟล์ประกอบการสอนของ อาจารย์ ปิยพล ยืนยงสถาวร เรื่อง : การเรียงลำดับข้อมูล