



ใบงานที่ 22

เรื่อง การเรียงลำดับข้อมูล

เสนอ

อาจารย์ ปิยพล ยืนยงสถาวร

จัดทำโดย

นาย สารินทร์ อินดีะรักษา รหัส 65543206082-1

ใบงานนี้เป็นส่วนหนึ่งของรายวิชา โครงสร้างข้อมูลและขั้นตอนวิธี

หลักสูตรวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา

ประจำภาคที่ 1 ปีการศึกษา 2566

คำสั่ง/คำชี้แจง

- สร้างโค้ดโปรแกรมตามตัวอย่างในเอกสารประกอบการสอน
- แสดงโค้ดโปรแกรมเป็นส่วนๆ พร้อมอธิบาย
- แสดงผลการรันโปรแกรม พร้อมอธิบายการทำงาน
- สรุปผลการทดลอง

ลำดับขั้นตอนการทดลอง

```
int Data[MaxData];
int Radix[MaxRow][MaxCol]; //Radix is temporary tank. The size is [0..MaxRow,0..MaxCol]
int N,N1;
void ClearStackPT() //Clear every block=0 and use Radix[0] for SP
{
    int i;
    for(i=0;i<=MaxRow;i++)
        Radix[i][0]=NULL; //This position keep the SP
}
void PrepareRawData(int N2)
{
    int i;
    srand(time(NULL)); //for difference random number in rand()
    for (i=0;i<N2;i++)
        Data[i]=(rand() % 899)+100; //random difference number 100..999
}
void DispData(int N2) //Show Data in array
{
    int i;
    for(i=0;i<N2;i++)
        printf("%3d ",Data[i]);
    printf("\n");
}
void Push(int Rad, int Dat) //Put data into Parallel Stack by keep SP at (Rad,0)
{
    int SP;
    SP=Radix[Rad][0]+1; //Skip SP
    Radix[Rad][0]=SP; //keep new SP
    Radix[Rad][SP]=Dat; //Push data into that Radix
}
```

- ประกาศตัวแปร Data เป็นอาร์เรย์ที่เก็บข้อมูลและ Radix เป็นอาร์เรย์สองมิติที่ถูกใช้เป็น "ถังชั่วคราว" (temporary tank) ในกระบวนการ Radix Sort.
- void ClearStackPT(): ฟังก์ชันนี้ใช้ในการล้างข้อมูลใน Radix โดยกำหนดให้ทุกค่าใน Radix ในคอลัมน์ที่ 0 (Radix[i][0]) เป็น NULL
- void PrepareRawData(int N2): ฟังก์ชันนี้ใช้ในการเตรียมข้อมูลสุ่ม โดยรับพารามิเตอร์ N2 เพื่อกำหนดจำนวนข้อมูลที่ต้องการ. ฟังก์ชันจะสร้างข้อมูลสุ่มในช่วง 100 ถึง 999 และเก็บไว้ในอาร์เรย์ Data.
- void DispData(int N2): ฟังก์ชันนี้ใช้ในการแสดงข้อมูลที่อยู่ในอาร์เรย์ Data. ฟังก์ชันนี้จะแสดงข้อมูลในรูปแบบของตัวเลขที่คั่นด้วยเว้นวรรค.
- ฟังก์ชัน Push ใช้ในการเพิ่มข้อมูล Dat ลงในถึง Rad โดยการเพิ่มค่าของ Stack Pointer (SP) และเก็บข้อมูลใน Radix ในตำแหน่งที่ SP ชี้ไป และทำการเพิ่มค่า SP ใน Radix ในคอลัมน์ที่ 0 เพื่อเก็บค่า SP ใหม่.

```

void ReadStack() //Read data from each Stack and transfer to Data[]
{
    int i,j,k,SP;
    k=0; //Start of Data[] at subscript 0
    for(i=0;i<=MaxRow;i++)
    {
        SP=Radix[i][0]; //Max Count value of each Radix keep at column 0
        for(j=1;j<=SP;j++)
        {
            Data[k]=Radix[i][j]; //Transfer data from every Stack to Data[]
            k++;
        }
    } //End for
} //End Fn.

```

- void ReadStack(): ฟังก์ชันนี้ใช้ในการอ่านข้อมูลจากแต่ละ Stack และโอนข้อมูลไปยังอาร์เรย์ Data.
- ทำการวนลูปตามรายการของ Radix (ถัง) ที่เก็บในอาร์เรย์ Radix.
- SP=Radix[i][0]: ในแต่ละ Radix, ฟังก์ชันนี้จะอ่านค่า Stack Pointer (SP) จาก Radix ในคอลัมน์ที่ 0 เพื่อระบุจำนวนข้อมูลใน Stack นี้.
- หลังจากได้ค่า SP แล้ว, ฟังก์ชันจะวนลูปตามค่าของ SP เพื่ออ่านข้อมูลจาก Stack นี้.
- ข้อมูลที่ถูกอ่านจาก Stack นี้จะถูกโอนไปยังอาร์เรย์ Data ในตำแหน่งที่เรียกว่า k.
- หลังจากที่มีข้อมูลถูกโอน, ค่า k จะถูกเพิ่มขึ้นเพื่อเลื่อนตำแหน่งที่ข้อมูลถูกเขียนลงในอาร์เรย์ Data.

```

void RadixSort(int N2)
{
    int Digit,i,RadixNo;
    char Txt[2];
    for(Digit=2;Digit>=0;Digit--) //Count down
    {
        printf("[Digit : %d]==>\n",3-Digit);
        for(i=0;i<N2;i++) //Data[] counter
        {
            itoa(Data[i],Txt,10); //convert Integer to Text [itoa(input,output,base)]
            RadixNo=Txt[Digit]-48; //convert Text that Digit to Number
            Push(RadixNo,Data[i]); //Push Data to Stack at that Radix
        } //End for
        ReadStack();
        DispData(N2);
        ClearStackPT();
    } //End for
} //End Fn.

```

- มีลูป for ที่เริ่มนับถอยหลังจากหลักหลักท้ายสุด (unit digit) ไปสู่หลักหลักแรก (hundreds digit)
- แสดงข้อความเพื่อรายงานถึงหลักที่กำลังใช้ในการ Radix Sort.
- มีลูป for อีกรอบที่วนตามอาร์เรย์ Data เพื่อคัดลอกข้อมูลมาทำการ Radix Sort.
- ในแต่ละรอบของลูป, ฟังก์ชัน itoa ใช้ในการแปลงค่าจำนวนเต็มใน Data[i] เป็นข้อความ (text) และเก็บผลลัพธ์ในอาร์เรย์ Txt.
- จากข้อความ Txt ที่ได้จากข้อมูลใน Data[i], ฟังก์ชันนี้คำนวณหาค่าของ Radix

- พร้อมข้อมูล Data[i] และค่า Radix RadixNo ที่คำนวณได้, ฟังก์ชันนี้จะนำข้อมูลไปใส่ Stack ที่เป็นของ Radix นั้น.
- หลังจากนำข้อมูลไปใส่ Stack ให้อ่านข้อมูลจาก Stack และโอนข้อมูลไปยังอาร์เรย์ Data
- ล้าง Stack โดยการกำหนดค่า SP ในทุกๆ Radix ในคอลัมน์ที่ 0 ให้เป็น NULL
- ฟังก์ชัน RadixSort จะทำ Radix Sort ในทุกหลัก (unit, ten, hundred) ในการเรียงลำดับข้อมูลทั้งหมด

```
int main()
{
    printf("ASCENDING RADIX SORT\n");
    printf("===== \n");
    N=16; //Change amount of N here
    N1=N; //Keep N to N1 Because N is 0 when finished RadixSort()!! I don't understand
    PrepareRawData(N);
    printf("Raw Data...\n");
    DispData(N);
    printf("----- \n");
    printf("Processing Data...\n");
    RadixSort(N);
    //printf("N= %d\n",N); //Problem point!!
    printf("-----Finished\n");
    printf("Sorted Data : \n");
    DispData(N1); //Sorted Data
    getch();
    return(0);
} //End Main
```

- "ASCENDING RADIX SORT" เพื่อรายงานว่าโปรแกรมจะทำการเรียงลำดับข้อมูลโดยใช้วิธี Radix Sort
- กำหนดค่า N เป็น 16, ซึ่งระบุจำนวนข้อมูลที่จะถูกเรียงลำดับ.
- ทำการคัดลอกค่า N ไปยัง N1 เพื่อให้สามารถเก็บค่าเริ่มต้นของ N ไว้ เนื่องจากค่า N จะถูกเปลี่ยนในฟังก์ชัน RadixSort.
- เรียกใช้ฟังก์ชัน PrepareRawData(N)
- แสดงข้อมูลที่อยู่ในอาร์เรย์ Data ในรูปแบบของตัวเลขที่คั่นด้วยเว้นวรรค.
- เรียกใช้ฟังก์ชัน RadixSort เพื่อทำการเรียงลำดับข้อมูล
- printf("-----Finished\n");
- แสดงข้อมูลที่ถูกเรียงลำดับในอาร์เรย์ Data ในรูปแบบของตัวเลขที่คั่นด้วยเว้นวรรค.
- getch();: ใช้รอผู้ใช้กดปุ่มคีย์ใดๆ บนแป้นพิมพ์เพื่อปิดหน้าต่างโปรแกรม
- return(0);: คืนค่า 0 เพื่อแสดงว่าโปรแกรมเสร็จสิ้น

ผลลัพธ์การทดลอง

```
C:\Users\Sarin\Desktop\ENG C  X  +  v

ASCENDING RADIX SORT
=====
Raw Data...
905 768 972 754 483 161 951 296 724 894 938 846 732 145 807 951
-----
Processing Data...
[Digit : 1]==>
161 951 951 972 732 483 754 724 894 905 145 296 846 807 768 938
[Digit : 2]==>
905 807 724 732 938 145 846 951 951 754 161 768 972 483 894 296
[Digit : 3]==>
145 161 296 483 724 732 754 768 807 846 894 905 938 951 951 972
-----Finished
Sorted Data :
145 161 296 483 724 732 754 768 807 846 894 905 938 951 951 972
```

สรุปผลการทดลอง

โปรแกรมนี้ใช้ในการสุ่มและเรียงลำดับข้อมูลตัวเลขใช้อัลกอริทึม RADIX SORT และแสดงผลลัพธ์ของการเรียงลำดับ

สื่อ / เอกสารอ้างอิง

ไฟล์ประกอบการสอนของ อาจารย์ ปิยพล ยืนยงสถาวร เรื่อง : การเรียงลำดับข้อมูล