

บทที่ 5

ผลการวิจัย

ผลการทำงานของโปรแกรมวัดความซับซ้อน เมื่อได้ทดลองนำโปรแกรมต่างๆมาทำการวัด ปรากฏว่าตัววัดแต่ละแบบสามารถบ่งบอกถึงประสิทธิภาพของโปรแกรมที่นำมาวัดได้ แม้ว่าตัววัดแต่ละแบบจะมีจุดที่นำมาพิจารณาหาความซับซ้อนจากโปรแกรมต่างกัน แต่ก็ให้ผลการทดสอบในแนวทางเดียวกันเป็นส่วนใหญ่ คือโปรแกรมที่มีค่าความซับซ้อนมากมักจะเป็นโปรแกรมที่มีประสิทธิภาพต่ำกว่าโปรแกรมที่ทำงานในลักษณะเดียวกันแต่มีค่าความซับซ้อนน้อยกว่า

ผลทดสอบการวัดความซับซ้อนโปรแกรม มีดังนี้

5.1 โปรแกรมทดสอบโปรแกรมที่ 1

ก. strcpy (char *s, char *t)

```
{
    while (1) {
        if (*t != '\0')
            return;
        *s = *t;
        s = s+1;
        t = t+1;
    }
}
```

ข. strcpy (char *s, char *t)

```
{
    while ((*s++ = *t++) != '\0');
}
```

วัดค่าความซับซ้อนของโปรแกรม ก. และ ข. ได้ดังนี้

5.1.1 โปรแกรม ก.

ตัววัดฮอลล์สตีค	ความยาว	23.51
	ปริมาตร	78.10
	ระดับของการโปรแกรม	0.10
	ระดับของภาษา	0.82
	ความล้าลึกของขั้นตอนกรรมวิธี	8.01
	ความพยายาม	1179.76
ตัววัด เซน	ค่าความซับซ้อน	4
ตัววัดแมคคูลู	ค่าความซับซ้อน	4
ตัววัดโอวิโด	ค่าความซับซ้อน	5

5.1.2 โปรแกรม ข.

ตัววัดฮอลล์สตีค	ความยาว	20.26
	ปริมาตร	64.24
	ระดับของการโปรแกรม	0.20
	ระดับของภาษา	2.57
	ความล้าลึกของขั้นตอนกรรมวิธี	12.85
	ความพยายาม	888.58
ตัววัด เซน	ค่าความซับซ้อน	2
ตัววัดแมคคูลู	ค่าความซับซ้อน	3
ตัววัดโอวิโด	ค่าความซับซ้อน	2

โปรแกรมทดสอบทั้ง ก. และ ข. เป็นการสำเนาค่าในตัวแปร t ไปยัง s จะพบว่าโปรแกรมทั้งสองแม้จะทำงานลักษณะเดียวกัน แต่ประสิทธิภาพของโปรแกรม ข. จะสูงกว่าโปรแกรม ก. ทั้งความเร็วและขนาดโปรแกรมที่เล็กกว่า และเมื่อพิจารณาจากค่าที่วัดได้จากตัววัดทั้งสี่แบบสามารถสรุปได้ดังนี้

1. ค่าความยาว, ปริมาตร และความพยายามของตัววัดฮอลล์สตีค ค่าจากตัววัด เซน ตัววัดแมคคูลู ตัววัดโอวิโด ของโปรแกรม ก. จะสูงกว่าโปรแกรม ข. แสดงว่าโปรแกรม ก. มีความซับซ้อนกว่าโปรแกรม ข. ซึ่งอาจหมายถึงว่าโปรแกรม ข. ดีกว่าโปรแกรม ก. ด้วย

2. จากตัววัดซอลสติค ค่าระดับของการโปรแกรม, ระดับของภาษา และ ความล้าสมัยของขั้นตอนกรรมวิธี ของโปรแกรม ก. ต่ำกว่าโปรแกรม ข. เนื่องจากค่าเหล่านี้จะเป็นสัดส่วนผกผันกับค่าปริมาตรของโปรแกรม คือปริมาตรลดลงค่าเหล่านี้จะเพิ่มขึ้น

5.2 โปรแกรมทดสอบโปรแกรมที่ 2

ก. char grade (int score)

```
{
    char gr;

    if (score >= 85) gr = 'a';
    if (score < 85 && score >= 75) gr = 'b';
    if (score < 75 && score >= 65 ) gr = 'c';
    if (score < 65 && score >= 50) gr = 'd';
    if (score < 50) gr = 'f';

    return gr;
}
```

ข. char grade (int score)

```
{
    if (score >= 85)
        return 'a';
    else if (score >= 75)
        return 'b';
    else if (score >= 65)
        return 'c';
    else if (score >= 50)
        return 'd';
    else
        return 'f';
}
```

วัดค่าความซับซ้อนของโปรแกรม ก. และ ข. ได้ดังนี้

5.2.1 โปรแกรม ก.

ตัววัดฮอลสตีค	ความยาว	49.66
	ปริมาตร	198.65
	ระดับของการโปรแกรม	0.15
	ระดับของภาษา	4.57
	ความล้าลึกของขั้นตอนกรรมวิธี	30.14
	ความพยายาม	5332.23
ตัววัดเซน	ค่าความซับซ้อน	2
ตัววัดแมคคลู	ค่าความซับซ้อน	9
ตัววัดโอวีโอด	ค่าความซับซ้อน	20

5.2.2 โปรแกรม ข.

ตัววัดฮอลสตีค	ความยาว	35.22
	ปริมาตร	126.26
	ระดับของการโปรแกรม	0.71
	ระดับของภาษา	64.42
	ความล้าลึกของขั้นตอนกรรมวิธี	90.18
	ความพยายาม	2223.38
ตัววัดเซน	ค่าความซับซ้อน	5
ตัววัดแมคคลู	ค่าความซับซ้อน	7
ตัววัดโอวีโอด	ค่าความซับซ้อน	8

p-3X

โปรแกรมที่นำมาทดสอบเป็นโปรแกรมการตัดเกรดทั้งสองโปรแกรม จะสังเกตเห็นได้ว่าโปรแกรม ก. มีวิธีการเขียนที่โปรแกรมที่ดีกว่าโปรแกรม ข. คือ มีการเปรียบเทียบและมีการใช้ตัวแปรน้อยกว่า ผลของการวัดปรากฏว่า โปรแกรม ก. มีค่าความซับซ้อนสูงกว่ายกเว้นตัววัดของเซน ซึ่งจากการทดสอบนี้อาจแสดงให้เห็นถึงจุดบกพร่องของตัววัดเซนได้ เพราะเซนจะคำนวณค่าความซับซ้อนจากจำนวนของจุดตัดที่มากที่สุด ระหว่างเส้นกับเส้นเชื่อมภายในกราฟกระแสการควบคุม ทำให้โปรแกรม ก. มีความซับซ้อนน้อยกว่า เนื่องจากคำสั่งควบคุม if ในโปรแกรม ก. ไม่เป็นคำสั่งที่ซ้อนกัน (nest if)

5.3 โปรแกรมทดสอบโปรแกรมที่ 3

```

ก. enum { FALSE, TRUE };
   int prime (int n)
   {
       int i;
       for (i=2; i<n; i++) {
           if (n % i == 0)
               return FALSE;
       }
       return TRUE;
   }

ข. enum { FALSE, TRUE };
   int prime (int n)
   {
       int i, bound;
       if (i % 2 == 0)
           return FALSE;
       if (i % 3 == 0)
           return FALSE;
       if (i % 5 == 0)
           return FALSE;
       bound = (int) sqrt ((float) n);
       for (i=7; i<bound; i=i+2) {
           if (n % i == 0)
               return FALSE;
       }
       return TRUE;
   }

```



วัดค่าความซับซ้อนของโปรแกรม ก. และ ข. ได้ดังนี้

5.3.1 โปรแกรม ก.

ตัววัดฮอลสตีค	ความยาว	23.50
	ปริมาตร	78.10
	ระดับของการโปรแกรม	0.13
	ระดับของภาษา	1.39
	ความล้าลึกของขั้นตอนกรรมวิธี	10.41
	ความพยายาม	1179.76
ตัววัดเซน	ค่าความซับซ้อน	3
ตัววัดแมคคลู	ค่าความซับซ้อน	5
ตัววัดโอวิโด	ค่าความซับซ้อน	5

5.3.2 โปรแกรม ข.

ตัววัดฮอลสตีค	ความยาว	48.00
	ปริมาตร	192.00
	ระดับของการโปรแกรม	0.08
	ระดับของภาษา	1.33
	ความล้าลึกของขั้นตอนกรรมวิธี	16.00
	ความพยายาม	5848.58
ตัววัดเซน	ค่าความซับซ้อน	3
ตัววัดแมคคลู	ค่าความซับซ้อน	8
ตัววัดโอวิโด	ค่าความซับซ้อน	14

โปรแกรมทั้งสองโปรแกรมที่ทำงานแบบเดียวกันคือหาค่าไพรม์ (prime) แต่โปรแกรม ข. ที่มีค่าความซับซ้อนมากกว่า จะทำงานได้เร็วกว่าโปรแกรม ก. ประมาณ 4 เท่า เมื่อ n มีค่าระหว่าง 0-1000 เนื่องจากโปรแกรม ข. มีการออกแบบที่ดี โดยให้มีการตรวจสอบค่าของข้อมูลในระหว่างการทำงาน เมื่อใดที่ข้อมูลมีค่าเกินขอบเขตที่ควรจะเป็นไปได้จะส่งค่ากลับไปยังโปรแกรมที่เรียก และเพิ่มตัวแปร bound เพื่อกำหนดขอบเขตของการวนซ้ำในคำสั่ง for ทำให้มีขนาดและมีคำสั่งควบคุมสายงานมากกว่า แต่จะมีการทำงานน้อยครั้งกว่า ผลจากการทดสอบนี้จะพบว่า บางครั้งแม้โปรแกรมจะมีความซับซ้อนน้อยกว่าก็ไม่ได้หมายความว่าโปรแกรมที่ดีกว่าเสมอไป อาจต้องเลือกระหว่างโปรแกรมที่มีขนาดใหญ่แต่มีการทำงานเร็ว หรือโปรแกรมขนาดเล็กแต่การทำงานช้า ขึ้นอยู่กับการนำไปใช้และประเภทของโปรแกรมนั้นๆ

5.4 โปรแกรมทดสอบโปรแกรมที่ 4

```
n. int find_access_type (char access[], int *type)
{
    struct TABLE {
        char access [4];
        int type;
    };

    struct TABLE ac_table [5] = {
        "GET", 1,
        "PUT", 2,
        "DEL", 3,
        "UPD", 4,
        "WRT", 5,
        "", -1,
    };

    int i;

    for (i=0; *ac_table [i].access; i++) {
        if (strcmp (ac_table [i].access, access)) {
            *type = ac_table [i].type;
            return SUCCESS;
        }
    }
    return FAIL;
}
```

```

๗. int find_access_type (char access[], int *type)
{
    *type = -1;
    if (!strcmp (access, "GET"))
        *type = 1;
    else
        if (!strcmp (access, "PUT"))
            *type = 2;
        else
            if (!strcmp (access, "DEL"))
                *type = 3;
            else
                if (!strcmp (access, "UPD"))
                    *type = 4;
                else
                    if (!strcmp (access, "WRT"))
                        *type = 5;

    return (*type == -1) ? FAIL : SUCCESS;
}

```

วัดค่าความซับซ้อนของโปรแกรม ก. และ ข. ได้ดังนี้

5.4.1 โปรแกรม ก.

ตัววัดฮอลล์สตีค	ความยาว	48.73
	ปริมาตร	194.92
	ระดับของการโปรแกรม	0.12
	ระดับของภาษา	2.76
	ความล้าลึกของขั้นตอนกรรมวิธี	23.20
	ความพยายาม	5290.85
ตัววัดเซน	ค่าความซับซ้อน	3
ตัววัดแมคคลู	ค่าความซับซ้อน	4
ตัววัดโอวีโอด	ค่าความซับซ้อน	6

5.4.2 โปรแกรม ข.

ตัววัดฮอลสตีค	ความยาว	66.58
	ปริมาตร	287.76
	ระดับของการโปรแกรม	0.06
	ระดับของภาษา	0.93
	ความล้าลึกของขั้นตอนกรรมวิธี	16.36
	ความพยายาม	11018.05
ตัววัด เซน	ค่าความซับซ้อน	6
ตัววัดแมคคลู	ค่าความซับซ้อน	6
ตัววัดโอวิโด	ค่าความซับซ้อน	22

โปรแกรมทดสอบที่ 4 เป็นการเปรียบเทียบโปรแกรมที่มีผลการทำงานเหมือนกัน ต่างกันที่โปรแกรม ก. จะมีการทำงานที่ไม่ซ้ำซ้อนเนื่องจากมีการเก็บค่าตัวแปรไว้ในตาราง แล้ววนลูปเปรียบเทียบ จึงทำงานเร็วกว่า และมีขนาดเล็กกว่า จากค่าที่วัดได้จะพบว่าโปรแกรม ก. ที่มีค่าความซับซ้อนน้อยกว่าโปรแกรม ข.

5.5 โปรแกรมทดสอบโปรแกรมที่ 5

```
ก. void get_header_msg (char head [])
{
    char temp [10];

    strcpy (head, " DATE : ");
    itoa (dd, temp, 10);
    strcat (head, temp);
    strcat (head, "/");
    itoa (mm, temp, 10);
    strcat (head, temp);
    strcat (head, "/");
    itoa (yy, temp, 10);
    strcat (head, temp);
    strcat (head, "/");
    strcat (head, " FROM : ");
    strcat (head, msg);
    strcat (head, " PAGE NO : ");
    itoa (page, temp, 10);
    strcat (head, temp);
}

ข. void get_header_msg (char head [])
{
    sprintf (head, " DATE : %d/%d/%d FROM : %s PAGE %d ",
            dd, mm, yy, msg, page);
}
```

วัดค่าความซับซ้อนของโปรแกรม ก. และ ข. ได้ดังนี้

5.5.1 โปรแกรม ก.

ตัววัดฮอลสตีค	ความยาว	48.18
	ปริมาตร	192.12
	ระดับของการโปรแกรม	0.02
	ระดับของภาษา	0.16
	ความล้าลึกของขั้นตอนกรรมวิธี	5.55
	ความพยายาม	5858.53
ตัววัด เซน	ค่าความซับซ้อน	1
ตัววัดแมคคลู	ค่าความซับซ้อน	0
ตัววัดโอวีโค	ค่าความซับซ้อน	1

5.5.2 โปรแกรม ข.

ตัววัดฮอลสตีค	ความยาว	31.26
	ปริมาตร	112.07
	ระดับของการโปรแกรม	0.13
	ระดับของภาษา	1.89
	ความล้าลึกของขั้นตอนกรรมวิธี	14.55
	ความพยายาม	2236.92
ตัววัด เซน	ค่าความซับซ้อน	1
ตัววัดแมคคลู	ค่าความซับซ้อน	0
ตัววัดโอวีโค	ค่าความซับซ้อน	1

โปรแกรมทดสอบที่ 5 เป็นการเปรียบเทียบโปรแกรมที่มีผลการทำงานเหมือนกัน แต่จะสังเกตได้ว่าโปรแกรม ข. จะสั้นกว่าโปรแกรม ก. มาก เนื่องจากมีการเลือกใช้คำสั่งที่เหมาะสมกว่า และเมื่อนำมาวัดความซับซ้อนพบว่าโปรแกรม ก. มีค่าความซับซ้อนมากกว่าโปรแกรม ข.