

FACULTY OF ENGINEERING
CHULALONGKORN UNIVERSITY
2110327 Algorithm Design

YEAR III, First Semester, Mid-term Examination, Aug 8, 2011, Time 8:00 – 11:30

ชื่อ-นามสกุล

เลขประจำตัว

										2	1
--	--	--	--	--	--	--	--	--	--	---	---

CR58

หมายเหตุ

1. ข้อสอบมีทั้งหมด 5 ข้อในกระดาษคำถามและใน website รวม จำนวน 11 หน้า คะแนนเต็ม 600 คะแนน
2. ไม่อนุญาตให้นำตำราและเครื่องคำนวณต่างๆ ใดๆ เข้าห้องสอบ
3. ควรเขียนตอบด้วยลายมือที่อ่านง่ายและชัดเจน
4. ห้ามการหยิบยื่นสิ่งใดๆ ทั้งสิ้น จากผู้สอบอื่นๆ เว้นแต่ผู้คุมสอบจะหยิบยื่นให้
5. ห้ามนำส่วนใดส่วนหนึ่งของข้อสอบออกจากห้องสอบ ข้อสอบเป็นทรัพย์สินของราชการซึ่งผู้ลักพาอาจมีโทษทางคดีอาญา
6. ผู้ที่ประสงค์จะออกจากห้องสอบก่อนหมดเวลาสอบ แต่ต้องไม่น้อยกว่า 45 นาที
7. เมื่อหมดเวลาสอบ ผู้เข้าสอบต้องหยุดการเขียนใดๆ ทั้งสิ้น
8. ผู้ที่ปฏิบัติเข้าข่ายทุจริตในการสอบ ตามประกาศคณะกรรมการวิศวกรรมศาสตร์
มีโทษ คือ ได้รับ สัญลักษณ์ F ในรายวิชาที่ทุจริต และพักการศึกษาอย่างน้อย 1 ภาคการศึกษา

รับทราบ

ลงชื่อนิสิต (.....)

หมายเหตุ (เพิ่มเติม)

1. โจทย์ทุกข้ออยู่ใน website <http://www.nattee.net/grader>
 - a. ข้อ 1 เป็นแบบปรนัย ให้เขียนคำตอบเป็นตัวอักษร ก,ข,ค, หรือ ง ลงในสมุดคำตอบหน้า 1
 - b. ข้อ 2 – 5 เป็นการเขียนโปรแกรมโดยใช้ระบบ grader (<http://www.nattee.net/grader>)
2. สำหรับข้อ 2 – 5 ถ้าไม่ต้องการตอบโดยใช้ grader นิสิตสามารถเลือกตอบลงในสมุดคำตอบได้ นิสิตสามารถตอบโดยเขียนบรรยายแนวคิดที่ implement ได้ในทางปฏิบัติ หรือจะเขียนเป็นรหัสเทียมประกอบแนวคิดที่นำเสนอด้วยก็ได้ **และต้องวิเคราะห์ประสิทธิภาพเชิงเวลาของอัลกอริทึมที่นำเสนอด้วย** นอกจากนี้ คะแนนที่ได้จะแปรตามประสิทธิภาพการทำงานของอัลกอริทึม
 - a. ถ้าต้องการเลือกตอบในสมุดคำตอบ ให้ทำเครื่องหมาย X ในข้อที่ต้องการด้านล่างนี้ และจะไม่มีการตรวจคำตอบใน grader เพิ่มเติม
 - b. การไม่ทำเครื่องหมาย X หมายความว่าให้คะแนนใน grader ซึ่งจะคิดจากการส่งที่ได้คะแนนมากที่สุดและจะไม่มีการตรวจคำตอบใน grader เพิ่มเติม
 - c. การตอบในกระดาษคำตอบจะไม่สามารถได้คะแนนมากกว่า 70% ของข้อดังกล่าว
3. การ login ด้วย account ผู้อื่นถือเป็นการทุจริตในการสอบ
4. ให้เขียนตอบข้อที่ k ไว้ที่หน้าที่ $2k - 1$ และ $2k$ ในสมุดคำตอบ ($k = 1, 2, 3, 4, 5$)
5. โจทย์แต่ละข้อจะมี code เริ่มต้นมาให้ นิสิตสามารถแก้ไข code ดังกล่าวอย่างไรก็ได้

- ข้าพเจ้าต้องการให้ตรวจข้อ 2 ในกระดาษคำตอบ
- ข้าพเจ้าต้องการให้ตรวจข้อ 3 ในกระดาษคำตอบ
- ข้าพเจ้าต้องการให้ตรวจข้อ 4 ในกระดาษคำตอบ
- ข้าพเจ้าต้องการให้ตรวจข้อ 5 ในกระดาษคำตอบ

1. ข้อสอบแบบปรนัย (200 คะแนน)

ให้ดูข้อสอบจาก website และให้เขียนคำตอบลงในสมุดคำตอบหน้า 1

2. ค่ามากที่สุดอันดับที่ 3 (3rd Largest Value) (100 คะแนน)

จงเขียนฟังก์ชันภาษา C หรือ C++ (จากโครงที่เขียนให้ข้างล่างนี้) เพื่อหาและคืนค่ามากที่สุดลำดับที่ 3 ของข้อมูลที่เก็บในอาร์เรย์ A (ขนาด n ช่อง)

```
int max3rd(int A[], int n) {  
    // A คืออาร์เรย์ที่มีขนาด n ตัวเลข  
    // ฟังก์ชันนี้ต้องคืนค่ามากที่สุดลำดับที่ 3 ใน A  
}
```

อาร์เรย์ A มีขนาดไม่น้อยกว่า 3 ตัว และข้อมูลใน A แตกต่างทั้งหมด การทำงานของฟังก์ชันนี้ควรมี time complexity เป็น $O(n \log n)$ (คำแนะนำ: น่าจะสามารถทำได้ในเวลา $O(n)$)

ตัวอย่าง

1. $A = \{1, 2, 3, 4, 5, 6, 7\}$ และ $n = 7$ ได้ค่ามากที่สุดลำดับที่สามเป็น 5
2. $A = \{5, 3, 1, 7, 8, 10\}$ และ $n = 6$ ได้ค่ามากที่สุดลำดับที่สามเป็น 7

เพิ่มโครงคำตอบ

เพิ่มโครงของคำตอบอยู่ในแฟ้มชื่อ max3rd.cpp นิสิตสามารถใช้แฟ้มนี้เป็นจุดเริ่มต้นได้ ภายในมีฟังก์ชัน main รับข้อมูลจากแป้นพิมพ์ เรียกใช้ฟังก์ชัน max3rd (ที่นิสิตต้องเขียน) และแสดงผลทางจอภาพแล้ว คุณไม่ควรแก้ไขฟังก์ชัน main

max3rd.cpp

```
#include <stdio.h>  
#include <stdlib.h>  
  
int max3rd(int A[],int n) {  
    int result;  
  
    // write your code here  
  
    return result;  
}  
  
int main(int argc, char **argv) {  
    int *A;
```

```

int i, n;
scanf("%d",&n);

// malloc A
A = (int*)malloc(sizeof(int) * n);

// read input
for (i = 0; i < n; i++) {
    scanf("%d", &A[i]);
}

printf("%d\n",max3rd(A,n));
}

```

ขอบเขตข้อมูลทดสอบ

สัดส่วนของข้อมูลทดสอบ	ลักษณะข้อมูลทดสอบ	ควรใช้เวลา
50%	$3 \leq n \leq 20,000$	$O(n^2)$
50%	$3 \leq n \leq 100,000$	$O(n \log n)$

ขอบเขตเวลา: 1 วินาที

ขอบเขตหน่วยความจำ: 32 MB

3. ผลรวมยกกำลังมอดุโล (Power Modulo Sum) (100 คะแนน)

ให้ A เป็นอาร์เรย์ของจำนวนเต็มขนาด m ช่อง จงเขียนฟังก์ชันภาษา C หรือ C++ (จากโครงที่เขียนให้ข้างล่างนี้) เพื่อหาและคืนค่าของ $\sum_{i=0}^{m-1} (A[i])^n \bmod k$ (ฟังก์ชันนี้ควรใช้เวลาเป็น $O(m \log n)$)

```

int modulo(int A[], int m, int n, int k) {
    // A คืออาร์เรย์ที่มีขนาด m จำนวนเต็ม
    // n คือเลขชี้กำลังที่ต้องการ และ k คือ modulo
    // ฟังก์ชันนี้ต้องคืนค่าของ  $\sum_{i=0}^{m-1} (A[i])^n \bmod k$ 
}

```

ตัวอย่าง

- ให้ $m = 1, n = 2, k = 3$ และ $A = \{ 5 \}$ ผลที่ได้คือ $5^2 \bmod 3 = 1$
- ให้ $m = 3, n = 2, k = 4$ และ $A = \{ 2, 5, 1 \}$ ผลที่ได้คือ $(2^2 \bmod 4) + (5^2 \bmod 4) + (1^2 \bmod 4) = 0 + 1 + 1 = 2$

เพิ่มโครงคำตอบ

เพิ่มโครงของคำตอบอยู่ในแฟ้มชื่อ modulo.cpp นิสิตสามารถใช้แฟ้มนี้เป็นจุดเริ่มต้นได้ ภายในมีฟังก์ชัน main รับข้อมูลจากแป้นพิมพ์ เรียกใช้ฟังก์ชัน modulo (ที่นิสิตต้องเขียน) และแสดงผลทางจอภาพแล้ว คุณไม่ควรแก้ไขฟังก์ชัน main

modulo.cpp

```
#include <stdio.h>
#include <stdlib.h>

int modulo(int A[], int m, int n, int k) {

    return 0;
}

int main(int argc, char **argv) {
    int *A;
    int i, n, m, k;
    scanf("%d %d %d",&m,&n,&k);

    // malloc A
    A = (int*)malloc(sizeof(int) * m);

    // read input
    for (i = 0; i < m; i++) {
        scanf("%d",&A[i]);
    }

    printf("%d\n",modulo(A,m,n,k));
}
```

ขอบเขตข้อมูลทดสอบ

สัดส่วนของข้อมูลทดสอบ	ลักษณะข้อมูลทดสอบ	ควรใช้เวลา
50%	$1 \leq n, m \leq 2,000$	$O(nm)$
50%	$1 \leq n, m \leq 10,000$	$O(n \log m)$

ขอบเขตเวลา: 1 วินาที

ขอบเขตหน่วยความจำ: 32 MB

4. ผลรวมตารางย่อยมากที่สุด (Maximum Subregion Sum) (100 คะแนน)

ให้ T คืออาร์เรย์สองมิติของจำนวนเต็มขนาด $n \times m$ ช่อง แต่ละช่องถูก indexed ด้วยคู่อันดับ $(0,0)$ ถึง $(n-1,m-1)$ ให้ “ตารางย่อย $(a,b) : (c,d)$ ของ T ” คือตารางย่อยของ T ที่ประกอบด้วยสมาชิกทุกตัวของ T ที่มีตำแหน่ง (x, y) ที่ $a \leq x \leq c$ และ $b \leq y \leq d$ และขออีกหนึ่งนิยาม ให้ “ตารางย่อยมากที่สุดของ T ” คือ ตารางย่อยของ T ที่ผลรวมของสมาชิกทั้งหมดมีค่ามากที่สุด

จงเขียนฟังก์ชันภาษา C หรือ C++ (จากโครงที่เขียนให้ข้างล่างนี้) เพื่อหาและคืนผลรวมของ “ตารางย่อยมากที่สุดของ T” (ฟังก์ชันนี้ควรใช้เวลาเป็น $O(nm)^2$)

```
int MSS(int T[100][100], int n, int m) {  
    // T คืออาร์เรย์ 2 มิติที่มีขนาด n x m  
    // ฟังก์ชันนี้ต้องคืนค่าของผลรวมของตารางย่อยมากที่สุดของ T  
}
```

ตัวอย่าง

1. สมมติให้ $n = 3, m = 4$ และตาราง T มีค่าเป็น

-1	-2	3	-2
4	3	5	-9
-3	8	9	2

ฟังก์ชันของคุณควรจะคืนค่า 26 (ซึ่งเกิดจากผลรวมของตารางย่อย (1,0) : (2,3))

2. สมมติให้ $n = 3, m = 3$ และตาราง T มีค่าเป็น

-2	-2	-2
-2	-1	-2
-2	-2	-2

ฟังก์ชันของคุณควรจะคืนค่า -1 (ซึ่งเกิดจากผลรวมของตารางย่อย (1,1) : (1,1))

เพิ่มโครงคำตอบ

เพิ่มโครงของคำตอบอยู่ในแฟ้มชื่อ mss.cpp นิสิตสามารถใช้แฟ้มนี้เป็นจุดเริ่มต้นได้ ภายในมีฟังก์ชัน main รับข้อมูลจากแป้นพิมพ์ เรียกใช้ฟังก์ชัน MSS (ที่นิสิตต้องเขียน) และแสดงผลทางจอภาพแล้ว คุณไม่ควรแก้ไขฟังก์ชัน main

mss.cpp

```
#include <stdio.h>  
#include <stdlib.h>  
  
int MSS(int T[100][100], int n, int m) {  
    int result = 0;  
  
    // this code simply show the table  
    // write your code here  
  
    // int i, j;  
    // for (i = 0; i < n; i++) {  
    //     for (j = 0; j < m; j++) {  
    //         printf("%4d", T[i][j]);  
    //     }  
    // }
```

```

// }
// printf("\n");
// }

return result;
}

int main(int argc, char **argv) {
    int T[100][100];
    int n,m,i,j;
    scanf("%d %d",&n,&m);

    // read input
    for (i = 0;i < n;i++) {
        for (j = 0;j < m;j++) {
            scanf("%d",&T[i][j]);
        }
    }

    printf("%d\n",MSS(T,n,m));
}

```

ขอบเขตข้อมูลทดสอบ

สัดส่วนของข้อมูลทดสอบ	ลักษณะข้อมูลทดสอบ	ควรใช้เวลา
50%	$1 \leq n,m \leq 20$	$O((nm)^3)$
50%	$1 \leq n,m \leq 100$	$O((nm)^2)$

ขอบเขตเวลา: 1 วินาที

ขอบเขตหน่วยความจำ: 32 MB

5. อันดับของลำดับย่อย (Order of Subset) (100 คะแนน)

ให้ $S(n)$ คือ string ของตัวอักษรภาษาอังกฤษเรียงตั้งแต่ a ถึงตัวที่ n ของตัวอักษรอังกฤษ (เช่น $S(1)$ คือ "a", $S(5)$ คือ "abcde", $S(26)$ คือ "abcdefghijklmnopqrstuvwxy") ดังนั้น n เป็นจำนวนเต็มบวกที่ไม่เกิน 26

ให้ลำดับย่อย (subsequence) ของ $S(n)$ มีนิยามเหมือนลำดับย่อยในปัญหา LCS (ถ้าไม่รู้วาลำดับย่อยของ string คืออะไร อ่านวงเล็บข้างล่างนี้)

(ลำดับย่อยของสตริง X คือ string ที่เกิดจากการลบตัวอักษรบางตัวออกจาก X โดยไม่มีการเปลี่ยนตำแหน่ง string ว่าง และ X เองก็ถือเป็นลำดับย่อยของ X ตัวอย่างเช่น "bce" เป็นลำดับย่อยของ $S(5)$)

ดังนั้น ลำดับย่อยทั้งหมดของ $S(3)$ ได้แก่ "", "a", "ab", "abc", "ac", "b", "bc", "c"

เราสามารถนำลำดับย่อยทุกแบบของ $S(n)$ มาเรียงตามอันดับอักษร (lexicographic order) ได้ เช่น

- “”, “a”, “ab”, “b” คือการเรียงตามอันดับอักษรของลำดับย่อยทั้งหมดของ S(2)
- “”, “a”, “ab”, “abc”, “ac”, “b”, “bc”, “c” คือการเรียงตามอันดับอักษรของลำดับย่อยทั้งหมดของ S(3)

ให้สังเกตจากตัวอย่างข้างบน จะพบว่า

- “b” อยู่เป็นลำดับที่ 4 ในการเรียงตามอันดับอักษรของลำดับย่อยทั้งหมดของ S(2)
- แต่ “b” อยู่เป็นลำดับที่ 6 ในการเรียงตามอันดับอักษรของลำดับย่อยทั้งหมดของ S(3)

ให้ w คือลำดับย่อยหนึ่งของ $S(n)$ จงเขียนฟังก์ชันภาษา C หรือ C++ (จากโครงที่เขียนให้ข้างล่างนี้) เพื่อหาและคืนว่า w ปรากฏเป็นลำดับที่เท่าใด ในการเรียงตามอันดับอักษรของลำดับย่อยทั้งหมดของ $S(n)$ (ฟังก์ชันนี้ควรใช้เวลาเป็น $O(n^2)$)

หมายเหตุ : เพื่อความสะดวกในการเขียนโปรแกรม เราจะ**ไม่ใช่ตัวอักษร** แต่จะใช้ตัวเลขแทนตัวอักษร ให้ 0 แทน ‘a’, 1 แทน ‘b’, ..., 26 แทน ‘z’ ดังนั้น w จะเป็นอาร์เรย์ของจำนวนเต็มที่มีขนาด k ตัว (k ก็คือความยาวของลำดับย่อย)

```
int order(int w[], int k, int n) {
    // w เป็น array มีความยาวเท่ากับ k ซึ่งเท่ากับจำนวนอักษรในลำดับย่อย
    // w[i] จะเก็บตัวเลขแทนอักษร โดยให้ 0 แทน a, 1 แทน b, ...
    // รับประกันว่า w ที่ได้รับเป็นลำดับย่อยหนึ่งของ S(n) แน่แน่นอน
    // n เป็นตัวแปรที่บอกถึง S(n) ที่เรากำลังสนใจ
    // ฟังก์ชันนี้คืนเลขลำดับที่ของ w ในการเรียงตามอันดับอักษรของลำดับย่อยทั้งหมดของ S(n)
    // ลำดับแรกคือ 1 ดังนั้น ถ้า k = 0 (แสดงว่าลำดับย่อยเป็นลำดับว่างๆ) คืน 1 แน่ ๆ
}
```

ตัวอย่าง

1. ให้ $n = 3$ และลำดับย่อย $w = \{0,1,2\}$, $k=3$ ฟังก์ชัน order ควรคืน 4 (เนื่องจาก “abc” ปรากฏเป็นลำดับที่ 4 การเรียงตามอันดับอักษรของลำดับย่อยทั้งหมดของ S(3))
2. ให้ $n = 4$ และลำดับย่อย $w = \{1,3\}$, $k = 2$ ฟังก์ชัน order ควรคืน 12 (เนื่องจาก “bd” ปรากฏเป็นลำดับที่ 12 ในการเรียงตามอันดับอักษรของลำดับย่อยทั้งหมดของ S(4))

เพิ่มโครงคำตอบ

เพิ่มโครงของคำตอบอยู่ในแฟ้มชื่อ order.cpp นิสิตสามารถใช้แฟ้มนี้เป็นจุดเริ่มต้นได้ ภายในมีฟังก์ชัน main รับข้อมูลจากแป้นพิมพ์ เรียกใช้ฟังก์ชัน order (ที่นิสิตต้องเขียน) และแสดงผลทางจอภาพแล้ว คุณไม่ควรแก้ไขฟังก์ชัน main

order.cpp

```
#include <stdio.h>
#include <stdlib.h>

int order(int w[], int k, int n) {
    // int i;
    // printf("%d %d\n", k, n);
    // for (i = 0; i < k; i++) {
```

```

// printf("%d ",w[i]);
// }
return 0;
}

int main(int argc, char **argv) {
    int w[100];
    int n, m, k;
    scanf("%d %d",&n, &m);
    int i, j;
    for (i = 0; i < m; i++) {
        scanf("%d",&k);
        for (j = 0; j < k; j++)
            scanf("%d",&w[j]);
        printf("%d\n",order(w,k,n));
    }
}

```

ขอบเขตข้อมูลทดสอบ

สัดส่วนของข้อมูลทดสอบ	ลักษณะข้อมูลทดสอบ	ควรรใช้เวลา
20%	$1 \leq n \leq 12$ และ $m \leq 1,000$	$O(2^n m)$
30%	$1 \leq n \leq 12$ และ $m \leq 10,000$	$O(2^n)$
50%	$1 \leq n \leq 26$ และ $m \leq 10,000$	$O(n^2 m)$

ขอบเขตเวลา: 1 วินาที

ขอบเขตหน่วยความจำ: 32 MB