คะแนน 70 (เก็บ 35)

Part 1 Choice 30 คะแนน

Part 2 ตอบคำถาม 25 คะแนน

Part 3 ตอบคำถาม 15 คะแนน

Part 1:

- ให้ x = {11,13,24,7} รวมกันให้ได้31 เซต S เป็นซับเซตของ x และ $s=\{x_i \mid x_i$ เป็นสมาชิกของ $(0,1)\}$ แล้วตอบคำถาม 3 ข้อ
    - มีปริภูมิสถานะทั้งหมดกี่เคสที่เป็นคำตอบ(2)
    - มีต้นไม้ปรภูมิสถานะที่โดนbacktrackกี่เคส(6)
    - เซต S คำตอบคืออะไร
- N-Queen ขนาด 4x4 มีจำนวนที่เป็นไปได้ทั้งหมดกี่วิธี
- Greedy ถอนเหรียญให้ได้ 37 มีเหรียญ 1, 5, 10
    - 10 10 5 5 1 1 1 1 1 1 1
    - 10 10 10 5 1 1 1
    - 1 1 1 1 1 1 1 1 5 5 10 10
- Huffman code ถามว่ามีกี่Byte
- ให้ต้นไม้Huffman มาและถามว่า AABEมีcodeคืออะไร(choiceเป็นbinary)
- ถามวิธีการของการทำHuffman encoding ว่าข้อไหนผิด
    - เรียงอักษรจากมากไปน้อย
    - เลือกจำนวนตัวอักษรที่มากที่สุด2คู่
    - ตรวยสอบชนิดของตัวอักษร
- ให้รูปgraphที่ระบายเส้นทึบมา และถามว่ามีโอกาศเป็นแบบไหนบ้าง
    - Prim
    - Kruskal
    - เป็นไปไม่ได้ทั้งคู่
    - เป็นทั้งคู่
- ตารางdynamic knapsack ตารางกำหนดมาแค่ค่าเริ่มต้น ถามM2,3 M4,5 มีค่าเท่าไหร่ และถามM5,6+M4,4
- Transitive closure โจทย์เหมือนที่จด และถามว่า d ไป c สั้นสุดเท่าไหร่
- ให้code N-Queen แล้วถามว่าตัวแปรนี้คืออะไรของ N-Queen และก็เลือกchoice เติมcode

- Activity selection เมื่อเลือกงานที่เสร็จก่อน จะต้องเปิดห้องกี่ห้อง และถ้ามีแค่2ห้องงานที่ทำได้มากที่สุดจะมีกี่งาน
- Knapsack ให้ตาราง มูลค่ากับน้ำหนัก และความจุกระเป๋า choice ไหนมีมูลค่ามากสุด(มีchoiceหลอกมูลค่ามากสุดแต่น้ำหนักเกิน)
- Dijkstra ทางที่สั้นที่สุดจากจุดหนึ่งไปอีกจุดหนึ่ง และถามว่าหลังจากทำวิธีนี้ครบแล้ว parent ของ node นี้คือใคร
- Prim ให้ทำวิธีนี้แล้วถามว่า path ไหนถูกต้อง (มันจะมีchoice pathที่เป็นloopดักอยู่)
- ให้สมการreccurrence relationที่เป็นdynamic มาสมมติ f(n) = f(n-1) + f(n-2) และถามว่า f(16) คือchoiceไหน

Part 2

- C(n,k) = C(n-1,k) + C(n-1, k-1) + C(n,k-1) ; C(0,0) = 2 C(n, 0) = 4 C(0, k) = 1 C(n, n) = 2 หา C(4,2)
- ให้code ฟังชันก์np() dynamic มาเป็นแบบ bottom up แล้วให้หาค่า np(6) (ให้หาสมการที่ดูจากcode)
- จากcode จงหาผลลัพท์ของ gen(ar[ ], 0, 4)

```
int gen(int x[], int k, int n)
{
    if (k == n)
    {
        int i;
        for (i = 0; i < n; i++)
        {
            if (x[i] == 1)
            {
                cout << ar[i] << " ";
            }
        }
        cout << endl;
    }
    else
    {
        x[k] = 0;
        gen(x, k + 1, n);
        x[k] = 1;
        gen(x, k + 1, n);
    }
}
```

- cho(n) = max(n+cho(n-2), cho(n-1); cho(n) = 1 เมื่อ n<=0 หา cho(6)
- หาf(8)

```
int f(int i)
{
    if(i == 0){
        return 1;
    }
    if(i == 1){
        return 2;
    }
    if(i == 2){
        return 3;
    }
    if(i == 3){
        return 4;
    }
    if(i == 4){
        return 5;
    }
    return f(i-1) + f(i-2) + f(i-3) + f(i-4) + f(i-5);

}
```

```
void calculator(int current, int target, int steps){
    if(current >= target){
        if(current == target){
            cout<<steps;
        }
    }
    calculator(current*2, target, steps+1);
    calculator(current+1, target, steps+1);
}
```

- วาดต้นไม้ปริภูมิสถานะของ calculator(2,9) อย่างน้อย 3 levels
- จงเขียนcode ที่มีการใช้backtracking และวาดต้นไม้ปริภูมิสถานะ อย่างน้อย3levels
- จงเขียนcode undirect graph หาvertexที่ไปได้ของแต่ละ vertex

Part 3

- เขียนcode Prim โดยใช้ vector(ไม่ได้ทำ)
- ให้สมการreccurrence relation เขียนcode dynamic bottom up โดยห้ามใช้ array(ไม่ได้ทำ)
- ให้สมการreccurrence relation มาและถามว่าแต่ละสมการถ้าใช้dynamic programming จะเหมาะสมหรือไม่ (ให้กากบาท ช่อง Yes No)