03603213: Algorithm Design and Analysis กำหนดส่ง 1/2/2023

Problem Set 6

ชื่อ: นางสาว ถิ่นธิชา รักสนิท **รหัสนิสิต:** 6330300054

1. ให้ directed acyclic graph $G = (V, E)$ และน้ำหนัก $w(u, v) \geq 0$ สำหรับแต่ละเส้นเชื่อม $(u, v) \in E$ จงออกแบบอัลกอริทึมที่หา path ที่มีน้ำหนักรวมมากที่สุดใน G พร้อมวิเคราะห์เวลาทำงาน
(คำแนะนำ: topological ordering)

Ans Psuedo code

```
def weight_DAG (vertex s):
    queue = new queue();
    vertex v, w;
    q.enqueue(s); // start vertex
    while (!q.Empty):
        v = q.Dequeue()
        for w ใน adjacent ถึง v:
            if(v.distance + cvw (cost weight) < w.distance):
                w.distance = v.distance + cvw;
                w.path     = v;
                if (w ไม่อยู่ใน q):
                    q.enqueue (w);
    return q.
```

O(n) { O(v) { ... }

Time complexity = O(nv)

2. (From Dasgupta,Papadimitriou,Vazirani) Give an $O(nt)$ algorithm for the following task.

   Input: A list of n positive integers $a_1, a_2, \ldots, a_n$; a positive integer t.
   Question: Does some subset of the a_i's add up to t? (You can use each a_i at most once.)

Ans ใช้ Dynamic programming แก้ปัญหา

Recursive Relative

Base case : $X(0,0) = \text{True}$

$X(0,x) = \text{False}$

if $a_i \leq j$

$X(j,i) = X(j-a_i, j-1)$ or $X(j,i-1)$

else:

$X(j,i) = X(j,i-1)$

Psuedo code

```
def subset():
  x(0,0)=1  // set true
  for i=1 ถึง n
    for j=1 ถึง t
      if a_i ≤ j:
        x(j,i) = x(j-a_i,j-1) or x(j,i-1)
      else :
        x(j,i) = x(j,i-1)
  return x(n,t)
```

Time complexity = $O(nt)$

3. (From Dasgupta,Papadimitriou,Vazirani) A *contiguous subsequence* of a list S is a subsequence made up of consecutive elements of S. For instance, if S is

$$5, 15, -30, 10, -5, 40, 10,$$

then $15, -30, 10$ is a contiguous subsequence but $5, 15, 40$ is not. Give a linear-time algorithm for the following task:

Input: A list of numbers, $a_1, a_2, \ldots, a_n$.
Output: The contiguous subsequence of maximum sum (a subsequence of length zero has sum zero).

For the preceding example, the answer would be $10, -5, 40, 10$, with a sum of 55. (Hint: For each $j \in \{1, 2, \ldots, n\}$, consider contiguous subsequence ending exactly at position j.)

Ans ใช้ Dynamic programming ช่วย

Recursive Relative

```
if i == 0
    s[i] = a
else
    s[i] = max(s[i-1] + a_i, a_i)
```

Psuedo code

```
def Subsequence():
    s[0] = 0
    check = 0
    for i = 1 ถึง n:
        s[i] = max(s[i-1] + a_i, a_i)
    for i = 0 ถึง n:
        if (s[i] > check):
            check = s[i]
    return check, s[]
```

Time complexity : O(n)