二、Answer **True** or **False** for the following statements and also justify your reasons.

(一) If an NP-complete problem can be reduced to a problem L in polynomial time, then L is NP-complete. (3%)

(二) If a problem A can be reduced to a problem B and $A \in P$, then $B \in P$. (3%)

(三) If a problem $A \in P$, then $A \in NP$. (3%)

(四) If $P \neq NP$, there exists a 2-approximation algorithm for the general traveling-salesman problem. (3%)

(五) The maximum numbers of elements in a heap of height h is $2^n - 1$. (3%)

參考解析 (一) False；(L 為 NP-hard 才對)

(二) Flase；(要是 polynomial-time reduced 才對)

(三) True； $(P \subseteq NP)$

(四) False；(原定理為：若 $P \neq NP$，則不存在 ρ -approximation algorithm 去解 general TSP，其中 $\rho \geq 1$)

(五) True；(即 full binary tree)

三、Given three strings $x[0, \ldots, n-1]$, $y[0, \ldots, m-1]$ and $z[0, \ldots, r-1]$. We say that z is a shuffle of x and y if it contains all characters of x and y and the left-to-right ordering of the characters from x and the characters from y is preserved. For example, "NcCKsiUe" is a shuffle of "NCKU" and "csie". The dynamic programming algorithm uses a tables S to check x, y and z, where S[i][j] is true if and only if the first $i + j$ characters of z are a shuffle of the first i characters of x together with the first j characters of y. Complete the following pseudocode for checks whether z is a shuffle of x and y by filling (1), (2) and (3). (please use C-style expression) (10%)

```
isShuffle(x, y, z)
    Let S[0…n][0…m] be a new table
    S[0][0] = true
    if r ≠ n + m
        return false
    for i = 1 to n
        S[i][0] = ________ (1) (3%) ________
    for j = 1 to m
        S[0][j] = ________ (2) (3%) ________
```

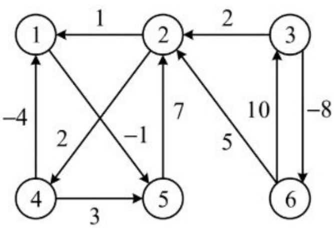
```
for  i = 1  to n
     for  j = 1  to m
          S[i][j] = ___________(3) (4%)___________
return  S[n][m]
```

參考解析 (一) $S[i-1][0] \&\& (z[i-1] == x[i-1])$

(二) $S[0][j-1] \&\& (z[j-1] == y[j-1])$

(三) $((z[i+j-1] == x[i-1]) \&\& S[i-1][j]) \| ((z[i+j-1] == y[j-1]) \&\& S[i][j-1])$

四、Consider the given directed graph.



The Floyd-Warshall algorithm can solve the all-pairs shortest-paths problem on a directed graph $G = (V, E)$. Answer the following questions.

(一) What is the time complexity of Floyd-Warshall algorithm? (2%)

(二) Let $d_{ij}^{(k)}$ be the weight of a shortest path from vertex i to vertex j for which all intermediate vertices are in the set $\{1, 2, \ldots, k\}$ and $D^{(k)} = (d_{ij}^{(k)})$ be a $n \times n$ matrix. Floyd-Warshall algorithm computes $D^{(k)}$ from $D^{(k-1)}$ as the following formula.

$d_{ij}^{(k)} =$ ______________

Please complete the above formula. (3%)

(三) Let dist(i, j) be the length of the shortest path from node i to node j. What is $\text{dist}(1, 5) + \text{dist}(2, 5) + \text{dist}(3, 5) + \text{dist}(4, 5) + \text{dist}(6, 5)$? (5%)

參考解析 (一) $\Theta(|V|^3)$

(二) $d_{ij}^{(k)} = \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$

(三) $\text{dist}(1, 5) = -1 \quad (V_1 \to V_5)$

$\text{dist}(2, 5) = -3 \quad (V_2 \to V_4 \to V_1 \to V_5)$

$\text{dist}(3, 5) = -6 \quad (V_3 \to V_6 \to V_2 \to V_4 \to V_1 \to V_5)$

Given three strings $x[0,\ldots,n-1]$, $y[0,\ldots,m-1]$ and $z[0,\ldots,r-1]$. We say that z is a shuffle of x and y if it contains all characters of x and y and the left-to-right ordering of the characters from x and the characters from y is preserved. For example, "NcCKsiUe" is a shuffle of "NCKU" and "csie". The dynamic programming algorithm uses a tables S to check x, y and z, where S[i][j] is true if and only if the first $i+j$ characters of z are a shuffle of the first i characters of x together with the first j characters of y. Complete the following pseudocode for checks whether z is a shuffle of x and y by filling (1), (2) and (3). (please use C-style expression) (10%)

題目: 給定 $X=\langle x_1,\ldots,x_n\rangle$ 為三 string，且 Z 包含所有 X, Y 字元且維持其 order

$Y=\langle y_1,\ldots,y_m\rangle$

$Z=\langle z_1,\ldots,z_r\rangle$

而 $S[i][j]$ 表示：當 z 前 $i+j$ 字元為 X 的前 i 且 Y 的前 j 字元之 shuffle 時為 true

Example: $X=\langle N, C, K, U\rangle$

$Y=\langle c, s, i, e\rangle$

$Z=\langle N, C, c, K, s, i, U, e\rangle$

DP 解:

① **Boundary Condition:** $S[0][0]=True$

$S[i][0]=[X[1,\ldots,i]==Z[1,\ldots,i]]\ \&\&\ s[i-1][0]$

$S[0][j]=[Y[1,\ldots,j]==Z[1,\ldots,j]]\ \&\&\ s[0][j-1]$

② **recursive relation:**

idea: $X=\langle N, C, K, U\rangle$

$Y=\langle c, s, i, e\rangle$

$Z=\langle N, C, c, K, s, i, U, e\rangle$

$S[3][2]=True \Rightarrow$ 此時 Z_{i+j} 必為 x_i 或 y_j

$S[3][2]=[(z_{i+j}==x_i)\ \&\&\ S[i-1,j]]\ ||\ [(Z_{i+j}==y_j)\ \&\&\ S[i,j-1]]$

操作題:

i \ j	0	1	2	3	4
0	T	F	F	F	F
1	T	F	F	F	F
2	T	T	F	F	F
3	F	T	T	T	F
4	F	F	F	T	T