A. 使用匈牙利演算法(Hungarian algorithm)來解以下的指派問題(assignment problem)（註: 必須寫出演算法執行過程中的每個中間結果）

|  |  |  |  |
| --- | --- | --- | --- |
|  | Task A | Task B | Task C |
| Tim | $1 | $2 | $3 |
| Bob | $2 | $3 | $2 |
| Alex | $2 | $2 | $3 |

step 1:每個row 扣除最小的值，並以最少線畫到所有0

|  |  |  |  |
| --- | --- | --- | --- |
|  | Task A | Task B | Task C |
| Tim | $0 | $1 | $2 |
| Bob | $0 | $1 | $0 |
| Alex | $0 | $0 | $1 |

Step2:每個人選出最適合的Task(以每個row或column只有一個0優先)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Task A | Task B | Task C |
| Tim | $0 | $1 | $2 |
| Bob | $0 | $1 | $0 |
| Alex | $0 | $0 | $1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Tim | 100 | 130 | 120 | 140 |
| Bob | 180 | 110 | 150 | 160 |
| Alex | 150 | 160 | 120 | 140 |
| Tony | 170 | 140 | 140 | 130 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Tim | 80 | 50 | 60 | 40 |
| Bob | 0 | 70 | 30 | 20 |
| Alex | 30 | 20 | 60 | 40 |
| Tony | 10 | 40 | 40 | 50 |

最小值:1+2+2 = 5  
B. 自行設計一個指派問題(assignment problem)的4x4成本矩陣，並以匈牙利演算法(Hungarian algorithm)找出最大權重完美二分匹配(Maximum-Weight Perfect Bipartite Matching)。  
step 1:轉換成最小化問題，用最大值-每個row的值  
  
  
  
  
  
step 2:每個row 扣除該row最小的值，並以最少線畫到所有0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Tim | 40 | 10 | 20 | 0 |
| Bob | 0 | 70 | 30 | 20 |
| Alex | 10 | 0 | 40 | 20 |
| Tony | 0 | 30 | 30 | 40 |

最少須三條，非最佳解  
  
step 3:每個column 扣除該最小的值，並以最少線畫到所有0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Tim | 40 | 10 | 0 | 0 |
| Bob | 0 | 70 | 10 | 20 |
| Alex | 10 | 0 | 20 | 20 |
| Tony | 0 | 30 | 10 | 40 |

最少須三條，非最佳解

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Tim | 50 | 10 | 0 | 0 |
| Bob | 0 | 60 | 0 | 10 |
| Alex | 20 | 0 | 20 | 20 |
| Tony | 0 | 20 | 0 | 30 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Tim | 50 | 10 | 0 | 0 |
| Bob | 0 | 60 | 0 | 10 |
| Alex | 20 | 0 | 20 | 20 |
| Tony | 0 | 20 | 0 | 30 |

step4:擴增元素(尚被覆蓋元素-尚被覆蓋元素的最小值)，兩條線覆蓋則加  
step5: 以最少線畫到所有0  
有四條，為最佳解  
  
step6:每個人選出最適合的(以每個row或column只有一個0優先)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Tim | 50 | 10 | 0 | 0 |
| Bob | 0 | 60 | 0 | 10 |
| Alex | 20 | 0 | 20 | 20 |
| Tony | 0 | 20 | 0 | 30 |

最大值:140+150+160+170 = 620

C. 說明如何使用匈牙利演算法(Hungarian algorithm)解決最小歐氏平面權重配對(Minimum Euclidean Weighted Matching)問題。所謂最小歐氏平面權重配對問題描述如下: 給定n個點(n為偶數)，如何將此n個點匹配形成n/2個點對，讓每個點對形成一條線段，而此n/2條線段具有最小的長度總和。  
step1:建立nxn的表格，表示從一點到任一點的距離，自己到自己為inf(因為無法配對)

|  |  |  |  |
| --- | --- | --- | --- |
|  | A | B | C |
| A | ∞ | 3 | 2 |
| B | 3 | ∞ | 1 |
| C | 2 | 1 | ∞ |

step2:利用匈牙利演算法，減去每個row、每個column最小值，以最少線畫到最到的零，直到以n條線畫到所有的0為止  
step3:找出每個row、column只有一個零，該格即為兩點配對的最小距離  
step4:因為圖為對稱，所以只須選一邊的三角形即可，被選中每個的原始值相加即為最小長度總和

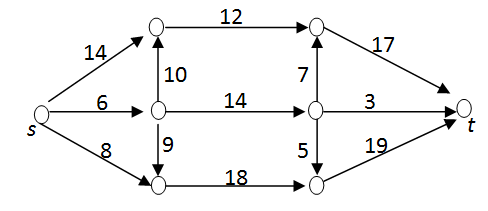
D. 隨意畫出四個歐氏平面點(2D point)，設其距離皆為整數(以公分計算)，以匈牙利演算法(Hungarian algorithm)找出其最小歐氏平面權重配對(Minimum Euclidean Weighted Matching)。  
step1:算出任一點到各點的距離

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| A | ∞ | 3 | 4 | 1 |
| B | 3 | ∞ | 2 | 4 |
| C | 4 | 2 | ∞ | 3 |
| D | 1 | 4 | 3 | ∞ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| A | ∞ | 2 | 3 | 0 |
| B | 1 | ∞ | 0 | 2 |
| C | 2 | 0 | ∞ | 1 |
| D | 0 | 3 | 2 | ∞ |

step2:以匈牙利演算法計算  
以4條線畫到所有零，找到最佳解  
  
  
  
  
step3: 找出每個row、column只有一個零，該格即為兩點配對的最小距離  
B match C、A match D

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| A | ∞ | 2 | 3 | 0 |
| B | 1 | ∞ | 0 | 2 |
| C | 2 | 0 | ∞ | 1 |
| D | 0 | 3 | 2 | ∞ |

E. 以Edmonds-Karp演算法解決以下最大流量問題(圖的有向邊(directed edge)上所標示的為容量(capacity))  
  
F. 自行 設計一個流網 (flow network)(除s、t外具有4個節點，並具有10個edge)，以Edmonds-Karp演算法解決 其最大流量問題