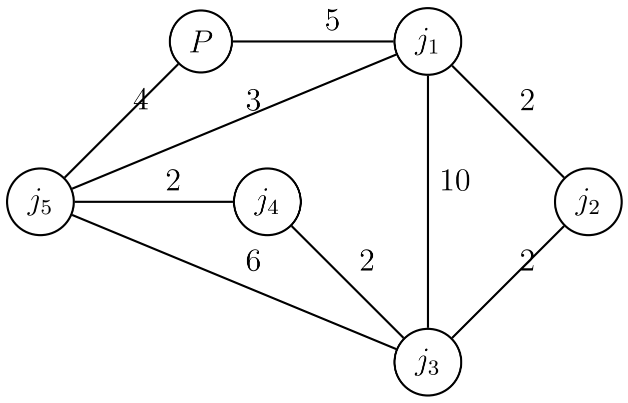
Week 8

Lecture 15 : Graph Theory[*]

8.1 Chinese Postman Problem

The Chinese Postman's dilemma is as follows: he wishes to travel along every road in a city in order to deliver letters in the shortest possible time. The problem is to determine the shortest closed walk through the graph that traverses each edge at least once rather than exactly once. A Euler cycle in a connected, weighted graph is referred to as the Chinese Postman problem in graph theory.

Example 1 :

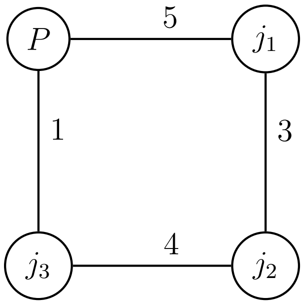


Case :

1. If every vertex and all vertices degree is even you can find the Euler circuit . For $(\forall v_1 \in V(G), degree(v_1) is even)$
2. If all edges are odd degree $(\exists v_1 \in V(G), degree(v_1) is odd)$

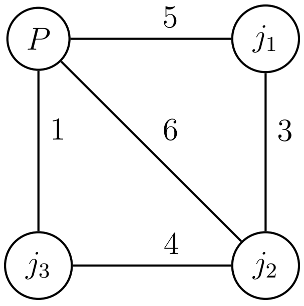
[*]Lecturer: Dr.Anand Misra. Scribe: Anggy Eka Pratiwi (P20CS002).

Solution Case 1 :

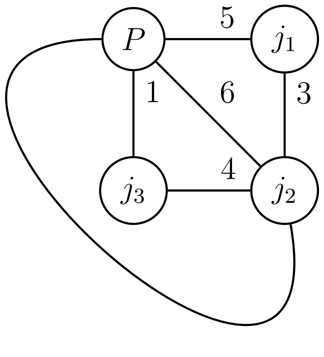


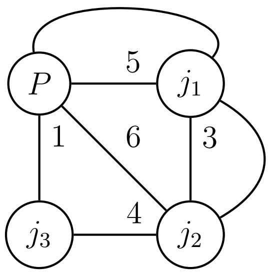
In the case 1 nothing to solve because of Euler circuit

Solution Case 2 :

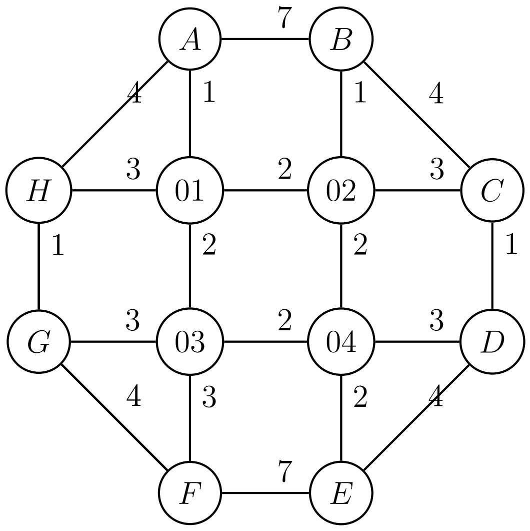


1. Euler Circuit does not exist in the case 2
2. You must visit node/edges twice in order to make Euler circuit
3. The solution is make every node even degree. The best solution is left one.

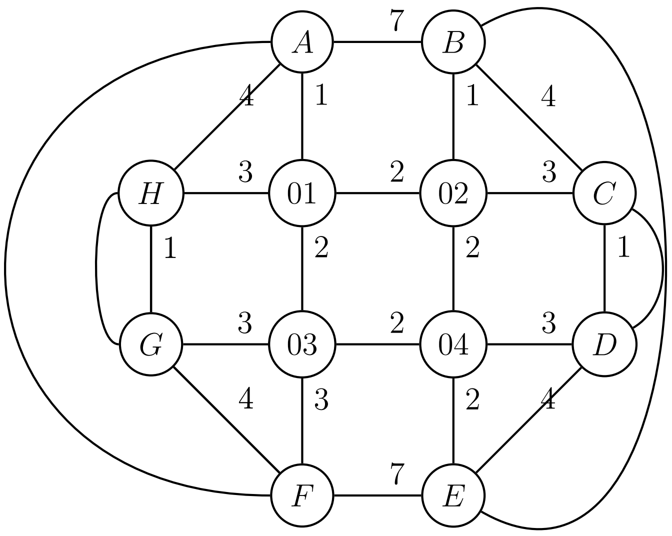




Example 2 :



Solution :
The example 2 is find the pair between HG, AF, CD, BE. and the minimum distance between the pair 4+4+1+1 = 10



8.2 Weighted Bipartite Matching

A Bipartite Weighted Graph is a graph that is G = (U,V,E) is a bipartite graph whose vertex set can be divided into two disjoint sets U and V such that each edge (ui,vj) ∈ E connects one vertex ui ∈ U and one vertex vj ∈V. If each edge in a graph G has an associated weight wij, the graph is said to be a weighted bipartite graph.

8.2.1 Construction Site problem

Case :

1. C : Train
2. S : Site
3. The problem is which train to send in which site, so it create minimum distance.

Example:

$$\begin{bmatrix} 5 & 10 & 15 & 20 \\ 3 & 5 & 6 & 9 \\ 9 & 8 & 1 & 15 \\ 15 & 16 & 1 & 5 \end{bmatrix}$$

From the example 1. rows for "Train" and Column for "Site"

8.2.2 Person Task Problem

Case :

1. P : Person
2. T : Task
3. The problem is to assign task in the correct in the correct person, So it can be finish fast.

Example:

$$\begin{bmatrix} 10 & 5 & 15 \\ 5 & 6 & 9 \\ 18 & 19 & 1 \end{bmatrix}$$

From the example above. rows for "Person" and Column for "Task"

8.2.3 Transversal

The term "graph traversal" (acronym for "graph search") refers to the process of inspecting (and/or updating) each vertex in a graph.A transversal of N x N matrix consist of N possitions one in each row and one in each column.

Example:
$$\begin{bmatrix} 11 & 12 & 13 \\ 21 & 22 & 23 \\ 31 & 32 & 33 \end{bmatrix}$$

Tranversal Solution

11	22	33
12	21	33
13	21	32

8.2.4 Assignment Problem as Optimization Problem

Condition : Minimize $\sum_j^i \sum_j^i C_{ij}\ X_{ij}$ S.T

$$C = \begin{bmatrix} Co_0 & Co_1 & Co_2 \\ -- & -- & -- \\ -- & -- & C_{nn} \end{bmatrix},$$

Case :
1. $\sum_j^i X_{ij} = 1$
2. $\sum_j^i (J_{ii}\ X_{ij} \in 0{,}1\ ,\ X_{ij} \geqslant 0$

8.3 Hungarian Algorithm

The Hungarian algorithm can be used to find a stable, maximum (or minimum) weight matching by manipulating the weights of the bipartite graph. This can be accomplished by determining a feasible labelling scheme for a graph that is perfectly matched, where a perfect match is defined as each vertex possessing exactly one matching edge.

$$\text{Cost Matrix} = \begin{bmatrix} 40 & 60 & 15 \\ 25 & 30 & 45 \\ 55 & 30 & 25 \end{bmatrix}$$

Algorithm :

1. Find row minimum
2. Subtract row minimum with every element

3. Find column min
4. Subtract column with min
5. if none of number required or = n so its complete

Solution :
Step 1 : Find row minimum

Row minimum = $\begin{bmatrix} 15 & 25 & 25 \end{bmatrix}$ from Cost Matrix = $\begin{bmatrix} 40 & 60 & 15 \\ 25 & 30 & 45 \\ 55 & 30 & 25 \end{bmatrix}$

Step 2 : Subtract row minimum with every element

Cost Matrix = $\begin{bmatrix} (40-15) & (60-25) & (15-15) \\ (25-25) & (30-25) & (45-25) \\ (55-25) & (30-25) & (25-25) \end{bmatrix}$ So the result C = $\begin{bmatrix} (25) & (45) & (0) \\ (0) & (5) & (20) \\ (30) & (5) & (0) \end{bmatrix}$

Step 3 : No line required in this problem so the process completed
Step 4 : Total cost optimal of the assignment is = 25+30+ 15 equal to 70

8.4 Time Complexity Analysis

Time Complexity: The time complexity of an algorithm quantifies the time required to run it as a function of the input length. Notably, the time required to run the algorithm is proportional to the length of the input, not to the actual execution time of the machine on which it is running.