* Traveling Salesperson Problem using Dynamic Programming:-

(Salesman or)

Dynamic Programming is that → try out all possible solution and pick up the best one.

→ Traveling problem consists of salesman and a set of cities. Here we have to find out the person who is travelling all the cities & the final ~~that~~ the cost of the cities is Minimum.

→ The salesman has to visit each city starting from Home and returning to the same city (i.e. Home)

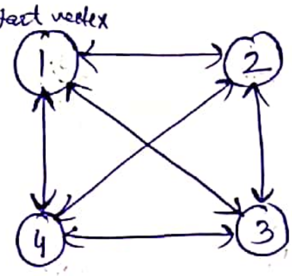
Main challenge :-

→ The person wants to minimize the total length of the trip. (or we say that minimum cost of travelling)

→ If visiting own city → the travelling cost is "Zero" i.e. starting and ending place ⇒ Cost is "Zero"

Example:- Weighted Adjacency Graph is Given

Start vertex



$$
\begin{array}{c c}
 & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\
A = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix}
\end{array}
$$

Cost of either direction may or may not be same. To avoid many parallel, directed uni-directional lines, we use bi-directional edges. Eg:- $1 \xrightarrow{10} 2$, $1 \xleftarrow{5} 2$

⇒ Formula:-

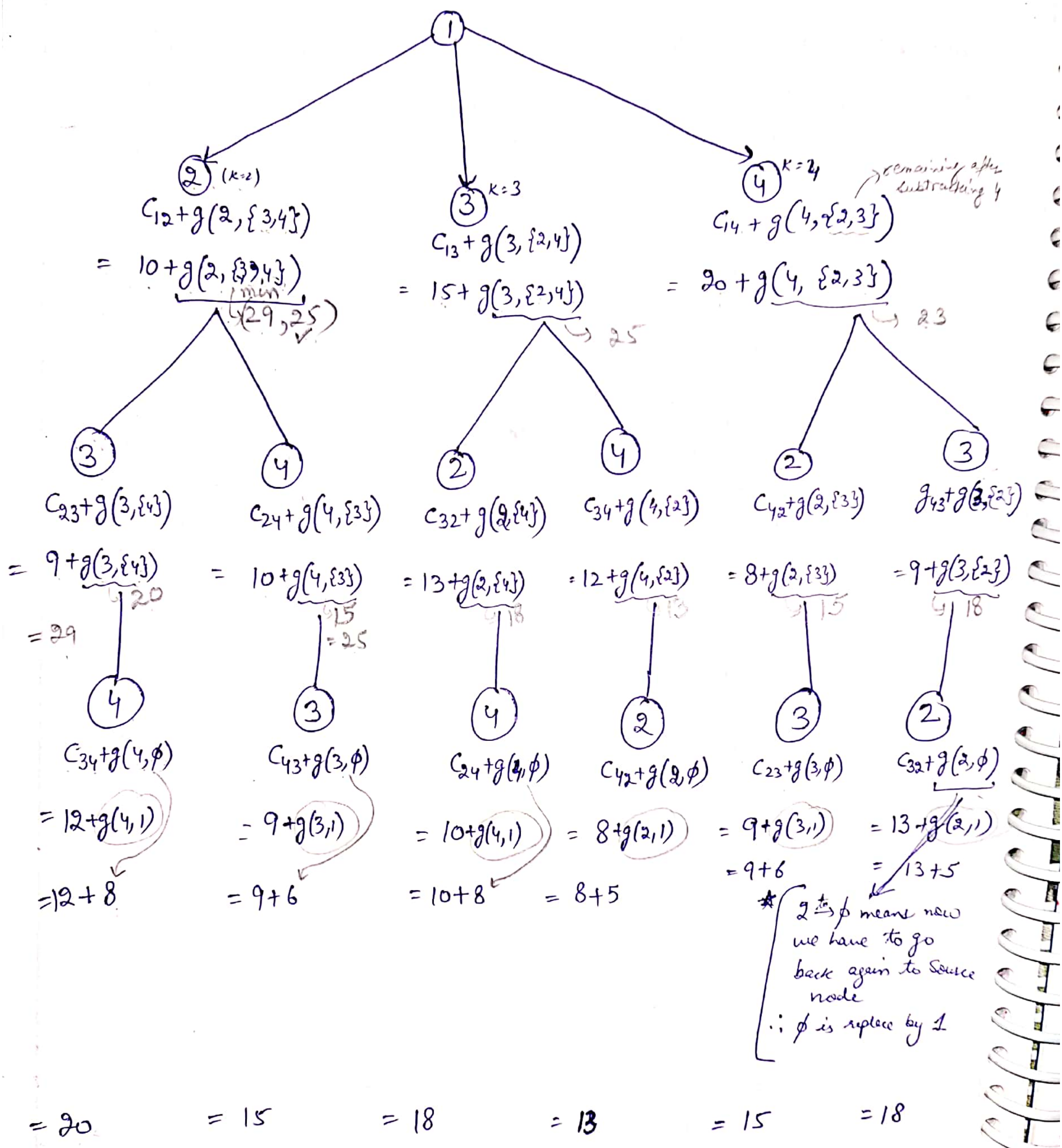
$$g(i,s) = \min_{K \in S} \left\{ C_{iK} + g(K, S-\{K\}) \right\}$$

Now Convert General formula into given Graph form:-

$$g(1, \{2,3,4\}) = \min_{K \in \{2,3,4\}} \left\{ C_{1K} + g\left(K, \{2,3,4\} - \{K\}\right) \right\}$$

$\boxed{K=2.}$ $C_{12} + g(2, \{3,4\})$.

Recursive Tree



Last level Nodes

$g(2, \phi) = 5$

$g(3, \phi) = 6$

$g(4, \phi) = 8$

2nd Last level Nodes

$g(2, \{3\}) = 15$

$g\{2, \{4\}) = 18$

$g(3, \{2\}) = 18$

$g(3, \{4\}) = 20$

$g(4, \{2\}) = 13$

$g(4, \{3\}) = 15$

2nd level Nodes

$g(2, \{3,4\}) = 25$

$g(3, \{2,4\}) = 25$

$g(4, \{2,3\}) = 23$

root node

$g(1, \{2,3,4\}) = 35$

So, The shortest route for Travelling Salesperson problem is = 35

Ans.