**7.6 Maximum Cut**

**Aim:** To construct a python code to solve the Maximum Cut problem.

**Algorithm:**

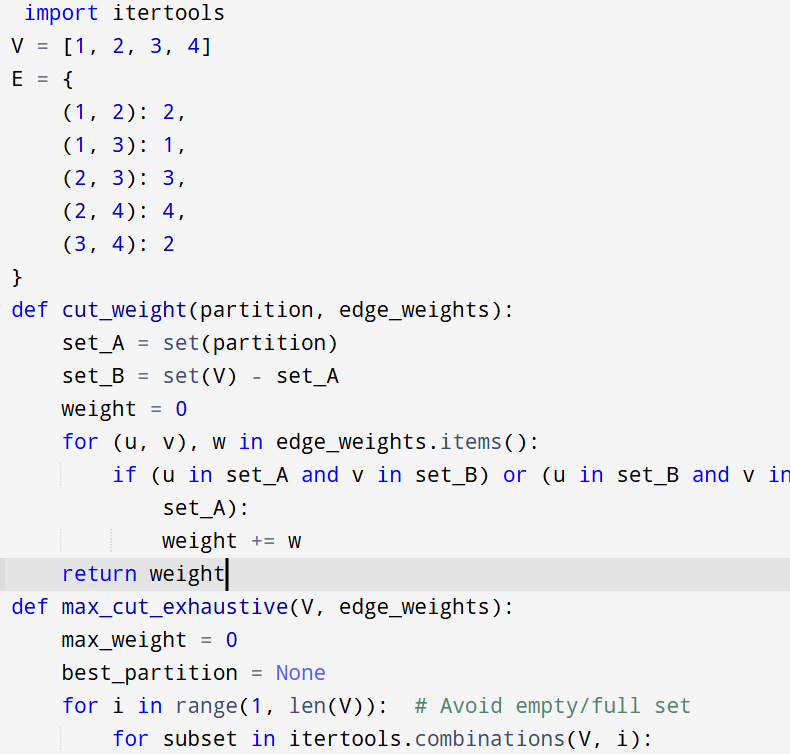
**🔹** Greedy Heuristic

* Start with random or empty partition.
* For each vertex, move it to the side that increases the cut weight the most.
* Stop when no improvement is possible.
* Time complexity: O(n × m)

🔹 Exhaustive Search (Brute-force)

* Enumerate all 2∣V∣2^{|V|}2∣V∣ possible partitions.
* Evaluate the cut weight for each.
* Return the partition with the maximum weight.
* Feasible for small graphs (like |V| = 4)
* Time complexity: O(2ⁿ × m)

**Program:**



**Input :**  V = [1, 2, 3, 4]

E = {

(1, 2): 2,

(1, 3): 1,

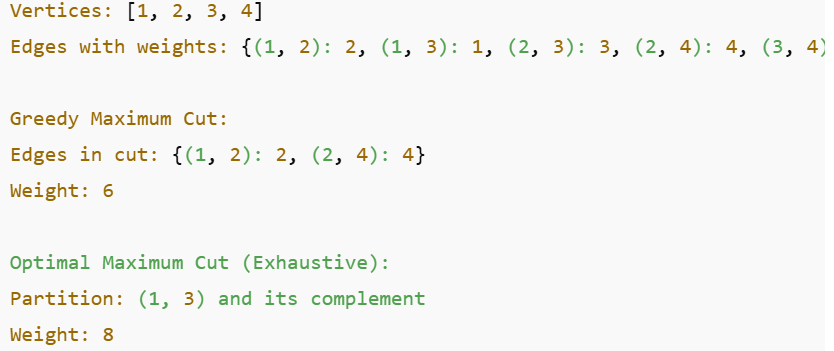
(2, 3): 3,

(2, 4): 4,

(3, 4): 2

}

**Output:**

****

**Result:** Program is been executed.

**Performance analysis:**

* Time complexity: O(n × m)
* Space complexity: O(n)