$S := \emptyset,\ T := \emptyset.$
for $v \in V$ **do**
    if $w(\{v\}, S) > w(\{v\}, T)$ **then**
        $T := T \cup \{v\}$
    else
        $S := S \cup \{v\}$
    end if
end for
return (S, T)

Figure 1: JohnsonCut

Show that JohnsonCut is an approximation algorithm for MAXCUT with an approximation ratio of 2.

In my answer, I use the notation $weight(p, v)$ *to denote the weight of edge* (p, v) *instead of* $w_{p,v}$ *for readability reason. The latter's subscript might be a little too small to read.*

Let (A, B) be the locally optimal cut. For every vertex v, the total weight of edges **incident** to this vertex v which **cross** the cut is at least the total weight of edges incident to v which **does not** cross the cut. This is the direct consequence of JohnsonCut algorithm.

Say, without loss of generality, that $v \in S$. So now we have

$$\sum_{p \in S} weight(p, v) \le \sum_{q \in T} weight(q, v)$$

where the LHS represents the total weight of edges incident to v and **does not** cross the cut, while the RHS represents the total weight of edges incident to v which cross the cut.

Summing over $v \in V$, for the LHS, is equivalent to summing over the weights of non-crossing edges twice. For example, a non-crossing edge (e,f) will be involved in $weight(e,f)$ and also $weight(f, e)$. Similarly, summing over $v \in V$, for the RHS, is equivalent to summing over the weights of crossing edges exactly twice.

Thus we have

$$2 * total\_weight(E_{non-crossing}) \le 2 * total\_weight(E_{crossing})$$

where $E_{non-crossing}$ is the set of all non-crossing edges and $E_{crossing}$ is the set of all crossing edges.

We now add $2 * total\_weight(E_{crossing})$ to both sides of the inequality:

$$2 * total\_weight(E_{non-crossing}) + 2 * total\_weight(E_{crossing}) \leq 2 * total\_weight(E_{crossing}) + 2 * weight(E_{crossing})$$

which becomes:

$$2 * total\_weight(E_{all}) \leq 4 * total\_weight(E_{crossing})$$

where E_{all} is the set of all edges.

Thus,

$$\frac{total\_weight(E_{all})}{total\_weight(E_{crossing})} \leq 2$$

The best possible maximum cut for a graph $G = (V, E)$ is where **all** of its edges $(i,j) \in E$ cross the cut. In other words, the optimal maxcut has weight $w(S, T) = total\_weight(E_{all})$ for partitions S and T where $V = S \cup T$ and $S \cap T = \emptyset$. Since this is a maximization problem, the approximation ratio is then

$$\frac{total\_weight(E_{all})}{total\_weight(E_{crossing})} \leq 2$$

It is shown that JohnsonCut is an approximation algorithm for MAXCUT with an approximation ratio of 2.

What can you say about the running time of the algorithm?

The algorithm essentially has two loops. The first loop, which is the outer loop, iterates through every vertex. This is bounded by the number of vertices.

For each vertex the algorithm iterates through the vertex's neighbours in order to compare the sum of weights of incident edges between those that cross the cut and those that do not. This is also bounded by the number of vertices, as the maximum number of neighbours a vertex can have is $|V| - 1$.

Therefore, JohnsonCut has a running time of $O(V^2)$.