2. Input: Matrix *A* of order m*n, vector *c* of length n, vector *b* of length m, a feasible point z of length n.
   Assume that rank of A is n, polytope ax<=b is non-degenerate.

Step 1 Moving to a vertex:

1. At point z divide rows of A into tight rows (A’) and un-tight rows (A’’).
2. *If the rank of tight rows (A’) is n:* then z is a vertex and we completed step 1.
3. *If the rank of tight rows (A’) is less than n*:
   - Take any non-zero vector u in the null space of tight rows (A’).
   - Find some positive/negative real number α such that at point *z+ αu* some row from un-tight rows (A’’) at z will be tight row.
   - Update z with z+ αu. Go to point 1.

Step 2: Vertex marching

1. At point z divide rows of A into tight rows (A’) and un-tight rows (A’’).
2. If cost vector c is a non-negative linear combination of tight rows (A’) then z is an optimum vertex.
3. If cost vector is not non-negative liner combination of tight rows (A’).
   - Compute inverse of A’. Let it be B.
   - Find a column v of B such that c.(-v) is positive.
   - Find some positive real number α such that at point *z- αv* some row from un-tight rows (A’’) at z will be tight row. (If you cannot find such a positive real number then cost function is unbounded and exit the algorithm by saying that cost is unbounded).
   - Update z with z- αv. Go to point 1.

Example:

-x - y <= -4
-x <=-1
-y <=-1

Maximize -5*x - 2*y