UNIVERSIDAD AUTONOMA DE NUEVO LEON.

FACULTAD DE INGENIERIA MECÁNICA Y ELECTRICA.

TAREA 2: ALGORITMO ASIGNACIÓN

PRESENTADO POR:

JESUS JAVIER MORENO VAZQUEZ 1619830

HORA: V4-V6

PROFESOR: DR. MARIA ANGELICA SALAZAR

SAN NICOLAS DE LOS GARZA, NUEVO LEON, A 30 DE ENERO DEL 2016

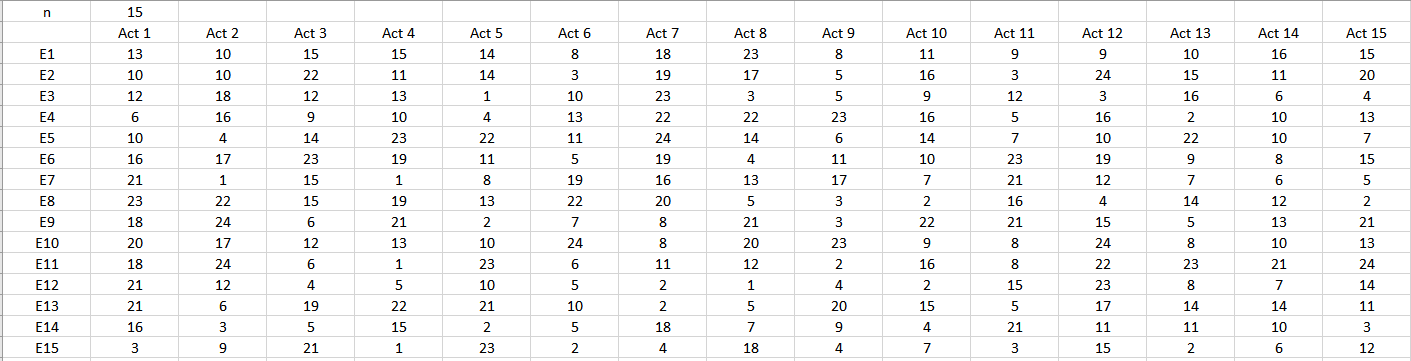
**DESCRIPCION GENERAL DEL PROBLEMA**

Se tiene una lista de tareas a repartir entre cierta cantidad de individuos que conforman un equipo, a cada individuo le toma cierto tiempo realizar cierta tarea, se necesita cubrir todas las tareas procurando que cada integrante del equipo tome 1 y solo 1 tarea con el menor tiempo posible. Se tienen 3 casos, uno correspondiente a las tareas de un equipo para desarrollar un proyecto, un caso sobre 15 empleados y 15 tareas y el tercer caso es similar al segundo, pero con algunas restricciones sobre 2 empleados.

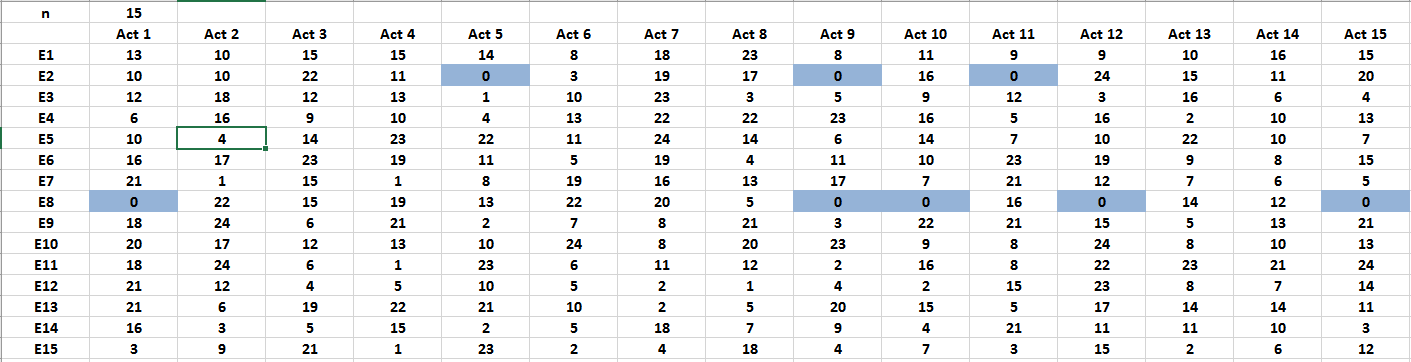
CASO 1: Propuesto en Clase

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Algoritmo | Implementación | Validación | Reporte | Presentación |
| Lucy | 12 | 20 | 35 | 28 | 15 |
| Jez | 4 | 10 | 10 | 2 | 8 |
| Tobías | 37 | 24 | 11 | 48 | 5 |
| Pedro | 1 | 7 | 23 | 32 | 6 |
| Javier | 15 | 37 | 40 | 43 | 8 |

CASO 2: Propuesto por docente



CASO 3: Propuesto por docente con restricciones en ciertos empleados



**ALGORITMO PROPUESTO**

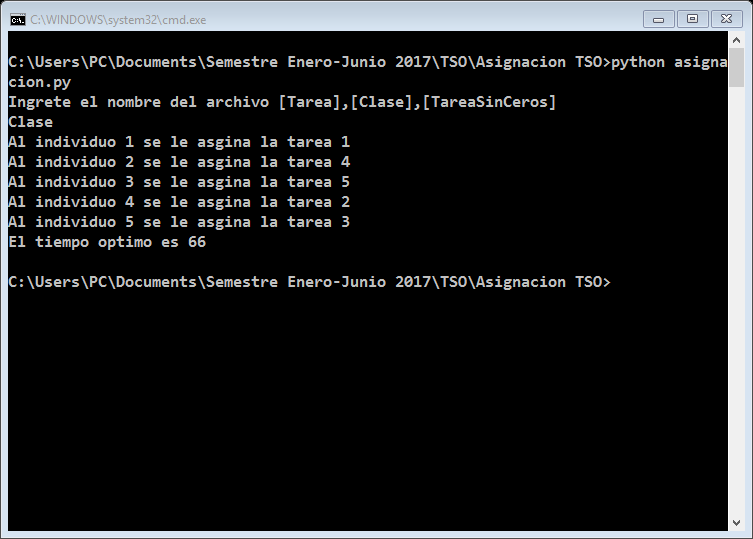
1. Lectura de datos
2. Revisa todos los datos para verificar que todos pueden realizar cualquier tarea
3. En caso de que algún individuo no pueda realizar alguna tarea, se reacomodaran los datos de tal forma que aquellos individuos con restricciones queden al principio de la matriz.
4. Se selecciona la tarea disponible que tome menos tiempo de para cada alumno en el orden dado
5. Se suman los tiempos de las tareas seleccionadas
6. Se reporta el resultado obtenido.

**MODIFICACIÓN**

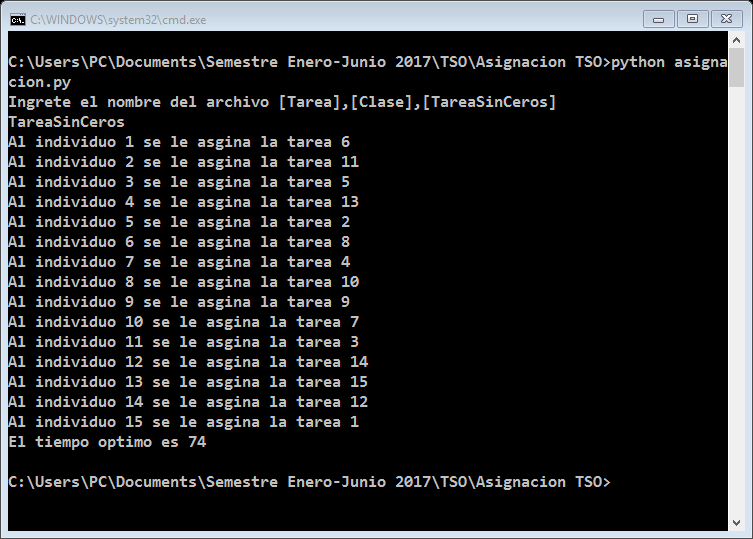
Para solucionar el caso propuesto en el que 2 empleados no pueden realizar ciertas tareas, lo que modifiqué en mi algoritmo fue reacomodar los datos de tal forma que dichos empleados queden al principio de la “matriz” para que sean los primeros en asignarse una tarea sin tomar en cuenta aquellas tareas que no pueden realizar.

**RESULTADOS**

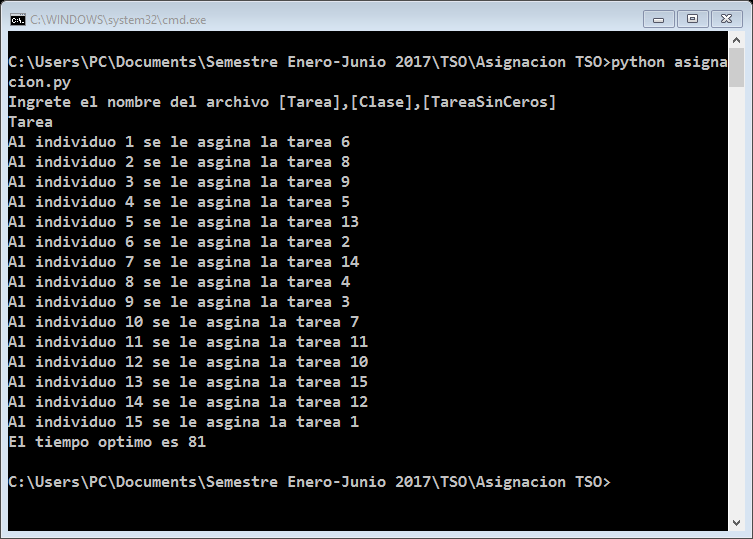
Caso 1:



Caso 2:



Caso 3:



\*Importante recordar que los individuos con restricciones pasaron a ser los individuos 1 y 2\*

**CONCLUSIONES**

Codificar este algoritmo de una manera correcta llevó más tiempo de lo previsto, al principio parecía muy sencillo de realizar, sin embargo, comenzaron a surgir ciertos errores de lógica que elevaron la dificultad de programar el algoritmo, principalmente al momento de reacomodar los datos, surgió un problema ya que en un ciclo yo intentaba buscar esos estudiantes con tareas que no pudiesen realizar y al momento de encontrarlo, sacarlo de esa lista y agregarlo a una nueva lista, sin embargo, al momento de sacar al estudiante el índice bajaba y el ciclo no podía terminarse por que se salía de su tamaño, un error de lógica que en lo personal fue tedioso de resolver. Yo recomiendo que se tome un tiempo considerable para la realización de tareas como esta, y una concentración total al momento de realizarlas.

**BIBLIOGRAFIA**

* https://www.python.org/doc/