EJEMPLOS Y CASOS DE EJECUCION CON RESULTADOS Y EXPLICACION DEL PROBLEMA ASIGNADO 3.3: TRABAJADORES Y TAREAS

Existe una carpeta "*CODIGOS Y EJECUTABLES*" en la que se encuentra un .cpp llamado "*trabajadores_tareas_greedy_por_pasos.cpp*" que no es un código precisamente eficiente, es una versión adaptada a que la salida por pantalla muestre paso por paso las comprobaciones y asignaciones que hace para facilitarlo al usuario que haga uso de él con el único fin de hacerlo comprensible. Una forma de probarlo sería compilarlo con el make o con la siguiente línea:

g++ trabajadores_tareas_greedy_por_pasos.cpp -o ejecutable

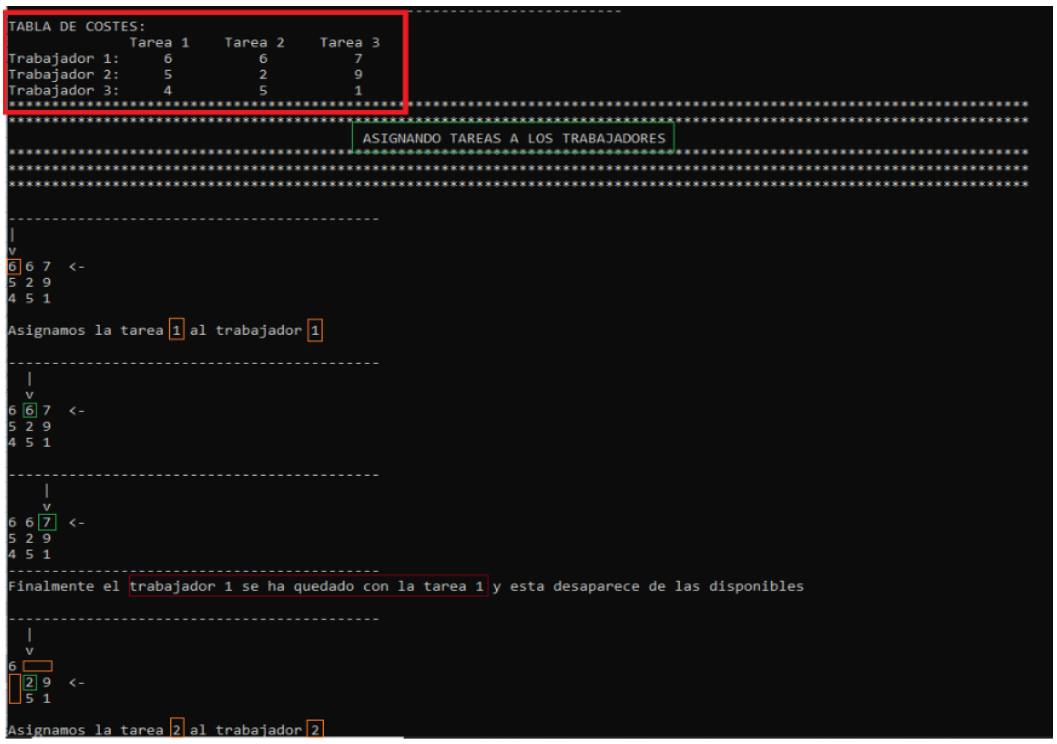
Y probarlo con la ejecución tal que:

./ejecutable [nº_tareas] [nº_trabajadores]

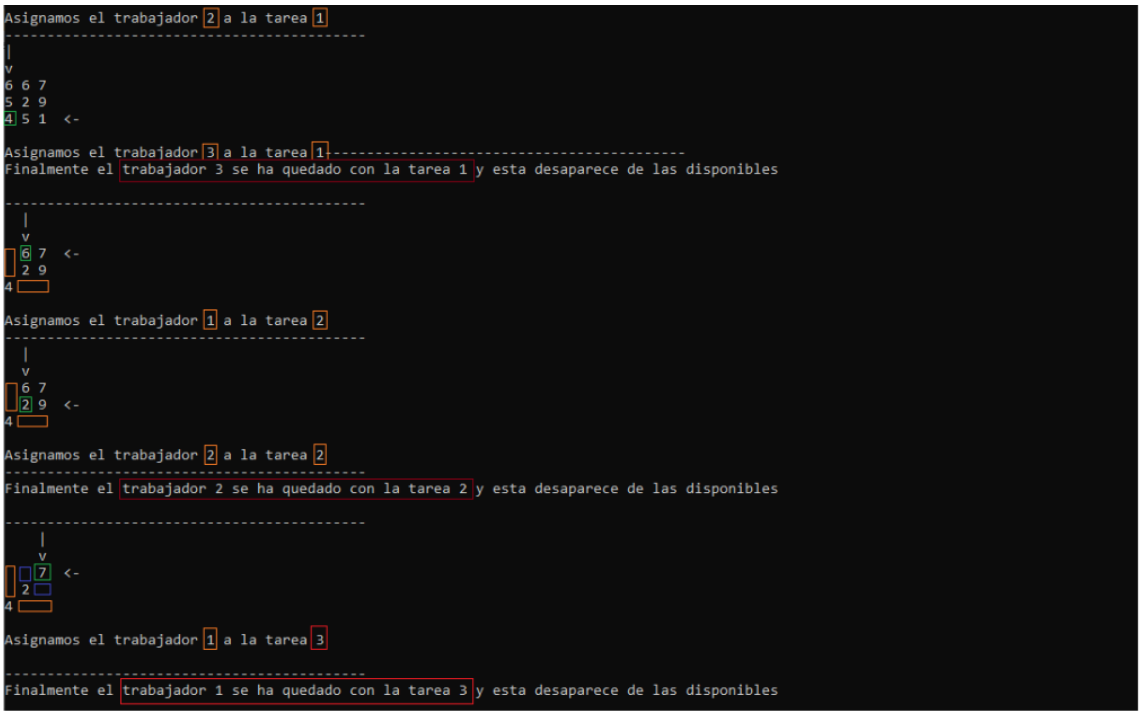
Para poder ver el ejemplo de cualquier caso que se considere oportuno. Para facilitar la comprobación, se adjuntan varios ejemplos que cubran las diferentes opciones de ejecución que pueden darse con el código y una breve explicación sobre estos.

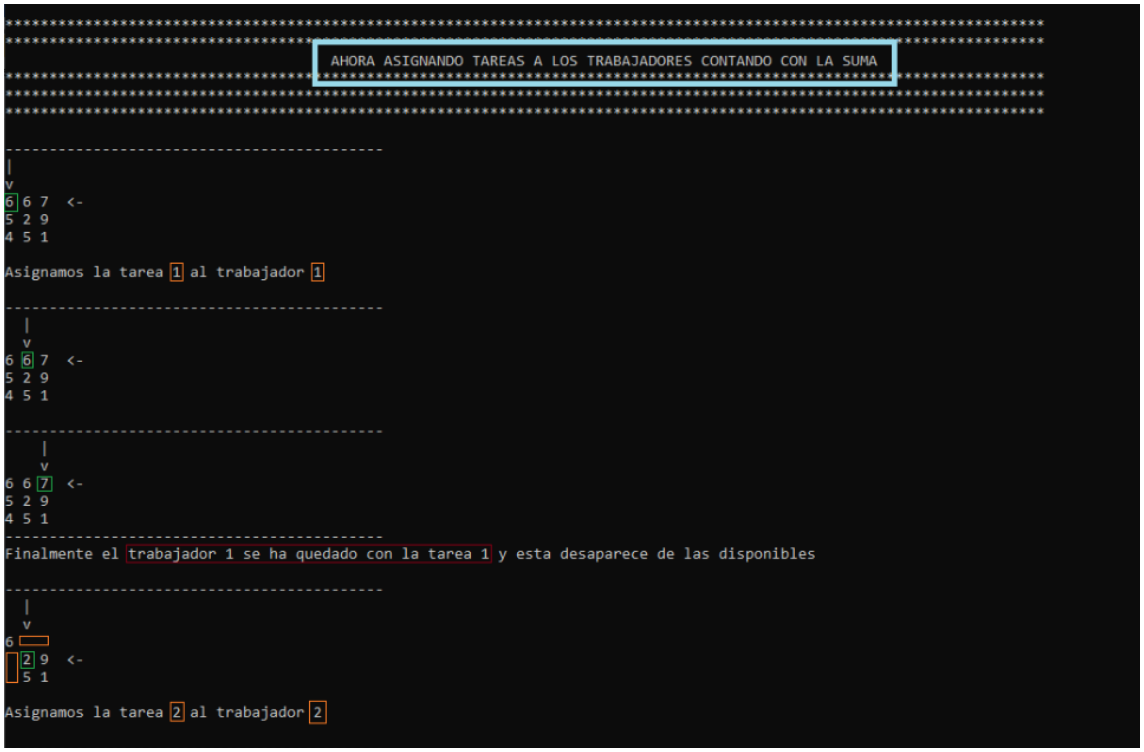
Ejemplo 1: Nº de trabajadores = 3 / Nº de tareas = 3. Ejecución estándar.

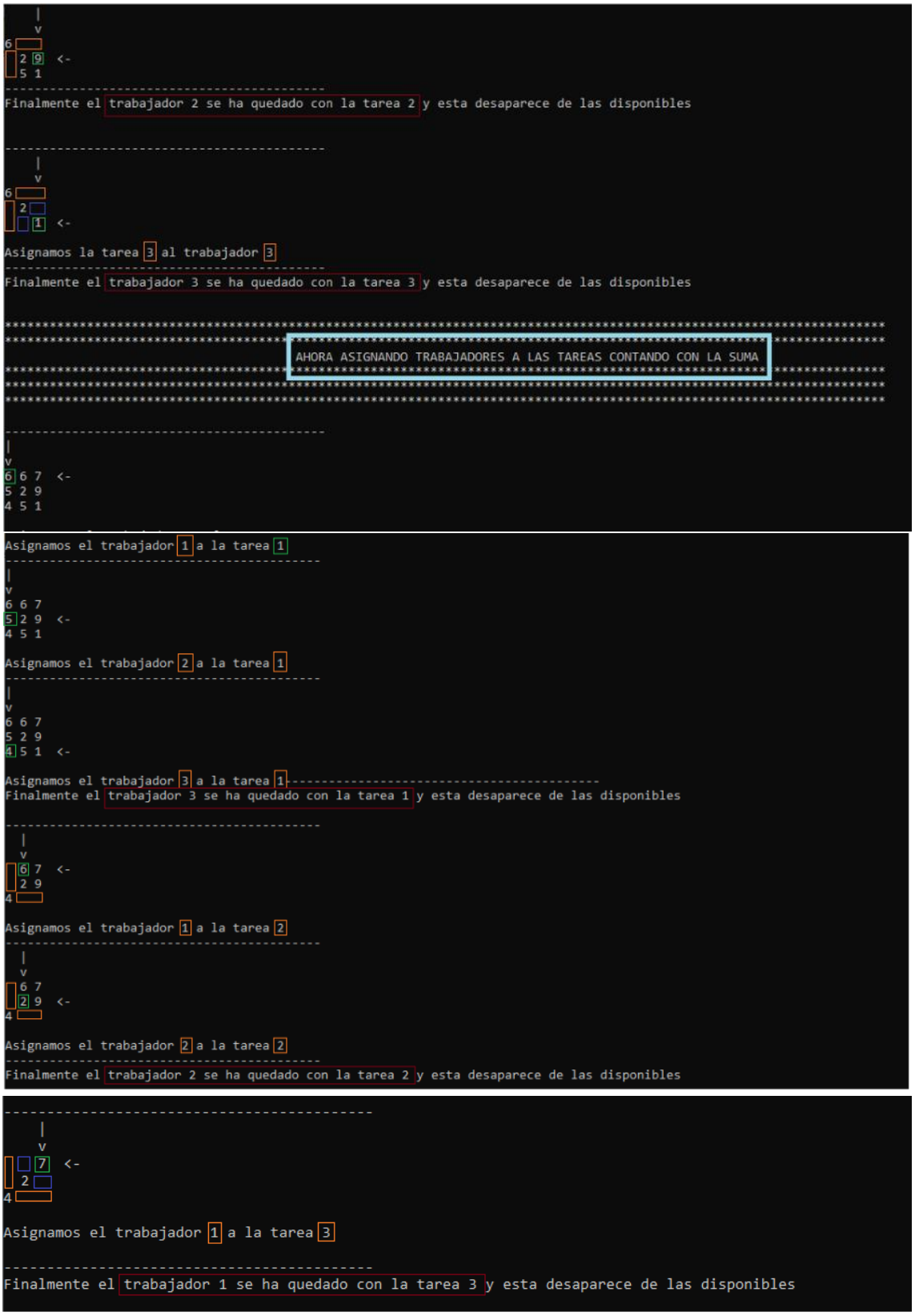
En este caso vamos a probar a introducir 3 trabajadores y 3 tareas. En el recuadro rojo se genera una tabla aleatoria de datos comprendidos en [1, 9] que representan los costes de la realización de las diferentes tareas (columnas) con los diferentes trabajadores (filas). Lo primero que se lleva a cabo es una asignación de tareas a trabajadores. Los recuadros verdes muestran qué elemento de la matriz se está estudiando. Estudiar un elemento consiste en comprobar si en esa fila o columna hay otra tarea o trabajador que cumpla con los requisitos necesarios y tenga un costo menor. Por defecto, la primera tarea disponible se asigna automáticamente al trabajador que se está evaluando y a raíz de eso ya depende del costo. Lo mismo ocurre con la asignación automática del primer trabajador libre para la tarea que se evalúe, facilitando las siguientes comprobaciones. Los cuadros naranjas son las asignaciones que se hacen en cada momento y los huecos que quedan en blando cuando se pasa de trabajador o tarea. Ocurre porque obligatoriamente una tarea es realizada por un trabajador solamente y un trabajador realiza una tarea solamente. Por tanto, al asignar un elemento, tanto la fila (el trabajador) como la columna (la tarea) dejan de estar disponibles para futuras asignaciones.



Las flechas que iteran indican qué elemento se va evaluando. La explicación del algoritmo se encuentra en un lugar externo a los ejemplos como este, pero básicamente comprueba si la tarea que se estudia tiene menos coste que otra asignada. De ser así, se le asigna esta y si no, es ignorada. Conociendo el funcionamiento básico, podemos ver a simple vista cómo avanzan los números a lo largo de las filas y columnas para hacer todas las comprobaciones. Si la relación trabajador-tarea que evalúa es mejor que la anterior, se muestra un texto bajo la matriz especificándolo y, al terminar la fila completa (haber evaluado para un trabajador i concreto todas las tareas j posibles), se indica finalmente qué trabajador se ha quedado qué tarea y se pasa a evaluar al siguiente (recuadro corinto).

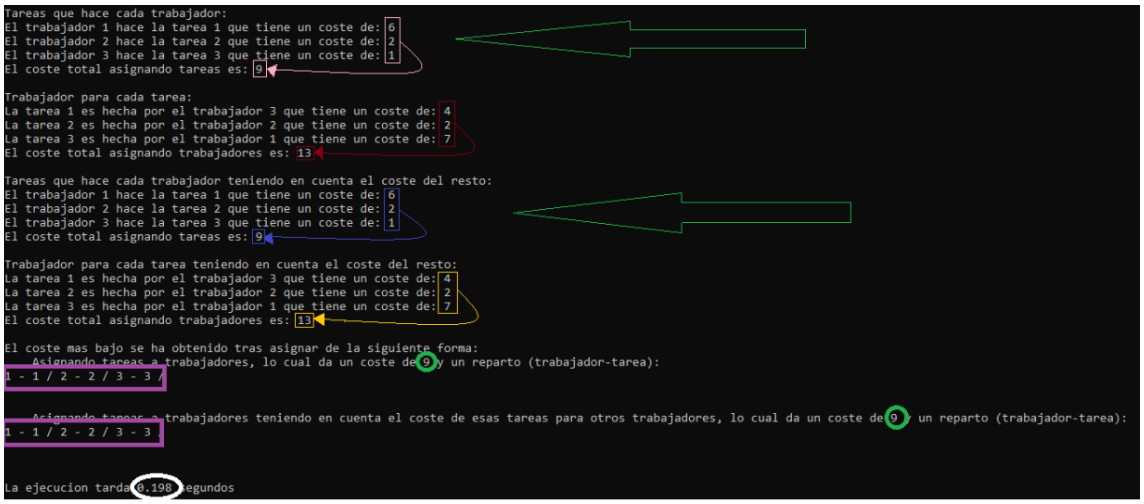

Asignamos el trabajador 2 a la tarea 1
6 6 7
5 2 9
4 5 1 <-
Asignamos el trabajador 3 a la tarea 1
Finalmente el trabajador 3 se ha quedado con la tarea 1 y esta desaparece de las disponibles
6 7 <-
2 9
4
Asignamos el trabajador 1 a la tarea 2
6 7
2 9 <-
4
Asignamos el trabajador 2 a la tarea 2
Finalmente el trabajador 2 se ha quedado con la tarea 2 y esta desaparece de las disponibles
7 <-
2
4
Asignamos el trabajador 1 a la tarea 3
Finalmente el trabajador 1 se ha quedado con la tarea 3 y esta desaparece de las disponibles


AHORA ASIGNANDO TAREAS A LOS TRABAJADORES CONTANDO CON LA SUMA
6 6 7 <-
5 2 9
4 5 1
Asignamos la tarea 1 al trabajador 1
6 6 7 <-
5 2 9
4 5 1
6 6 7 <-
5 2 9
4 5 1
Finalmente el trabajador 1 se ha quedado con la tarea 1 y esta desaparece de las disponibles
6
2 9 <-
5 1
Asignamos la tarea 2 al trabajador 2



Los huecos vacíos tras cada comprobación resaltan la ausencia de valores en la fila y columna que acaban de ser emparejadas como una asignación definitiva. De esta forma, y con el

objetivo de limpiar la salida para el usuario, se elimina la muestra de estos valores sucesivamente con cada línea que evaluamos.



En la anterior imagen se muestra por pantalla el resultado de ambos estudios realizados. Un pequeño resumen que recopila qué tareas son realizadas por qué trabajadores en cada caso y una suma de sus costes (colores rosa, rojo, azul y amarillo). A partir de la obtención de datos, se muestra una respuesta coherente con los valores obtenidos. En este caso concreto, como se ve con las flechas verdes, sale mejor resultado asignando tareas a trabajadores y tareas a trabajadores con un razonamiento algo más específico para el caso de lo empates de coste. Como el desarrollo de la asignación acaba eliminando toda situación de empate, el 3º algoritmo y el primero realizan el mismo procedimiento y obtienen las mismas asignaciones, pero más adelante veremos que no siempre es así y existen casos para los que razonar los empates es mejor opción. Finalmente, muestra la relación trabajadores-tareas que se produce como mejor. Otro dato observable es el tiempo que tarda **solamente el algoritmo que estudia las asignaciones**, no el tiempo de muestra por pantalla o recepción de datos. En este caso, al ser un código alterado, el tiempo no es totalmente válido, por lo que es mejor medir este en la ejecución estándar del fichero o en el fichero adjunto “trabajadores_tareas_greedy_tiempos.cpp”.

Tras esta explicación, el resto de ejemplos serán expuestos como una sucesión de imágenes que muestran el mismo procedimiento ya observado, pero para diferentes casos, haciéndonos ver diferentes alternativas de ejecución y resultados.

Ejemplo 2: Nº de trabajadores = 4 / Nº de tareas = 4. Igual al ejemplo anterior, pero con otras dimensiones.

```
------------------------------------------------------------------
TABLA DE COSTES:
             Tarea 1    Tarea 2    Tarea 3    Tarea 4
Trabajador 1:     6          3          7          5
Trabajador 2:     1          1          7          9
Trabajador 3:     4          1          2          5
Trabajador 4:     6          1          3          8
**********************************************************************************************************************
**********************************************************************************************************************
                                        ASIGNANDO TAREAS A LOS TRABAJADORES
**********************************************************************************************************************
**********************************************************************************************************************
**********************************************************************************************************************


-----------------------------------------
|
v
6 3 7 5  <-
1 1 7 9
4 1 2 5
6 1 3 8

Asignamos la tarea 1 al trabajador 1

-----------------------------------------
  |
  v
6 3 7 5  <-
1 1 7 9
4 1 2 5
6 1 3 8

Asignamos la tarea 2 al trabajador 1

-----------------------------------------
    |
    v
6 3 7 5  <-
1 1 7 9
4 1 2 5
6 1 3 8

-----------------------------------------
      |
      v
6 3 7 5  <-
1 1 7 9
4 1 2 5
6 1 3 8
-----------------------------------------
Finalmente el trabajador 1 se ha quedado con la tarea 2 y esta desaparece de las disponibles
-----------------------------------------
|
v
  3
1   7 9  <-
4   2 5
6   3 8

Asignamos la tarea 1 al trabajador 2


-----------------------------------------
    |
    v
  3
1   7 9  <-
4   2 5
6   3 8

-----------------------------------------
      |
      v
  3
1   7 9  <-
4   2 5
6   3 8
-----------------------------------------
Finalmente el trabajador 2 se ha quedado con la tarea 1 y esta desaparece de las disponibles


-----------------------------------------
```

```
------------------------------------------
    |
    v
  3
1
    2 5  <-
    3 8

Asignamos la tarea 3 al trabajador 3

------------------------------------------
      |
      v
  3
1
    2 5  <-
    3 8
------------------------------------------
Finalmente el trabajador 3 se ha quedado con la tarea 3 y esta desaparece de las disponibles



------------------------------------------
      |
      v
  3
1
    2
      8  <-

Asignamos la tarea 4 al trabajador 4
------------------------------------------
Finalmente el trabajador 4 se ha quedado con la tarea 4 y esta desaparece de las disponibles


**********************************************************************************************************************
**********************************************************************************************************************
                                        AHORA ASIGNANDO TRABAJADORES A LAS TAREAS
**********************************************************************************************************************
**********************************************************************************************************************
**********************************************************************************************************************

------------------------------------------
------------------------------------------
|
v
6 3 7 5  <-
1 1 7 9
4 1 2 5
6 1 3 8

Asignamos el trabajador 1 a la tarea 1
------------------------------------------
|
v
6 3 7 5
1 1 7 9  <-
4 1 2 5
6 1 3 8

Asignamos el trabajador 2 a la tarea 1
------------------------------------------
|
v
6 3 7 5
1 1 7 9
4 1 2 5  <-
6 1 3 8

------------------------------------------
|
v
6 3 7 5
1 1 7 9
4 1 2 5
6 1 3 8  <-
------------------------------------------
Finalmente el trabajador 2 se ha quedado con la tarea 1 y esta desaparece de las disponibles

------------------------------------------
```

En la siguiente imagen se han añadido algunos colores a un caso concreto para facilitar el entendimiento de lo que ocurre con los huecos en blanco, los números que quedan y las comprobaciones por realizar. Para hacerlo más visual:

- Cuadros rosas: Asignaciones ya hechas. El coste de los emparejamientos que van hasta ahora, colocados en la fila y columna que corresponde con su trabajador y tarea correspondiente. Obviamente, con la restricción de que una trabajador solo realice una tarea y que una tarea sea únicamente realizada por un trabajador, cuando se muestran datos como los recuadrados en rosa, se observa que no coinciden ni en fila ni en columna con ningún otro dato.
- Huecos naranjas: Números que se han eliminado tras la primera asignación (la tarea 1 con el trabajador 2 y coste 1)
- Huecos azules: Números que se han eliminado tras la segunda asignación (tarea 2, trabajador 3 y coste 1)
- Números en rojo: Coste de tareas aún asignables en las filas de trabajadores que aún pueden tener asignaciones nuevas.
- Número en verde: Evaluado en ese momento (se ve por las flechas de alrededor de la matriz)

```
  |
  v
  3 7 5  <-
1
  1 2 5
  1 3 8

Asignamos el trabajador 1 a la tarea 2

-----------------------------------------
  |
  v
  3 7 5
1
  1 2 5  <-
  1 3 8

Asignamos el trabajador 3 a la tarea 2
-----------------------------------------
  |
  v
  3 7 5
1
  1 2 5
  1 3 8  <-
-----------------------------------------
Finalmente el trabajador 3 se ha quedado con la tarea 2 y esta desaparece de las disponibles

-----------------------------------------
    |
    v
  7 5  <-
1
  1
  3 8

Asignamos el trabajador 1 a la tarea 3
```

FIGURA DE EJEMPLO

```
-----------------------------------------
    |
    v
    7 5
1
  1
    3 8  <-

Asignamos el trabajador 4 a la tarea 3-----------------------------------------
Finalmente el trabajador 4 se ha quedado con la tarea 3 y esta desaparece de las disponibles

-----------------------------------------
     |
     v
     5  <-
1
  1
    3

Asignamos el trabajador 1 a la tarea 4


-----------------------------------------
Finalmente el trabajador 1 se ha quedado con la tarea 4 y esta desaparece de las disponibles



**********************************************************************************************************************
**********************************************************************************************************************
                                    AHORA ASIGNANDO TAREAS A LOS TRABAJADORES CONTANDO CON LA SUMA
**********************************************************************************************************************
**********************************************************************************************************************
**********************************************************************************************************************

-----------------------------------------
|
v
6 3 7 5  <-
1 1 7 9
4 1 2 5
6 1 3 8

Asignamos la tarea 1 al trabajador 1

-----------------------------------------
```

```
----------------------------------------
  |
  v
6 3 7 5  <-
1 1 7 9
4 1 2 5
6 1 3 8

Asignamos la tarea 2 al trabajador 1

----------------------------------------
    |
    v
6 3 7 5  <-
1 1 7 9
4 1 2 5
6 1 3 8

----------------------------------------
      |
      v
6 3 7 5  <-
1 1 7 9
4 1 2 5
6 1 3 8
----------------------------------------
Finalmente el trabajador 1 se ha quedado con la tarea 2 y esta desaparece de las disponibles
----------------------------------------
|
v
  3
1   7 9  <-
4   2 5
6   3 8

Asignamos la tarea 1 al trabajador 2


----------------------------------------
```

```
----------------------------------------
    |
    v
  3
1   7 9  <-
4   2 5
6   3 8

----------------------------------------
      |
      v
  3
1   7 9  <-
4   2 5
6   3 8
----------------------------------------
Finalmente el trabajador 2 se ha quedado con la tarea 1 y esta desaparece de las disponibles


----------------------------------------
    |
    v
  3
1
    2 5  <-
    3 8

Asignamos la tarea 3 al trabajador 3

----------------------------------------
      |
      v
  3
1
    2 5  <-
    3 8
----------------------------------------
Finalmente el trabajador 3 se ha quedado con la tarea 3 y esta desaparece de las disponibles
```

```
----------------------------------------
     |
     v
  3
1
   2
     8  <-

Asignamos la tarea 4 al trabajador 4
----------------------------------------
Finalmente el trabajador 4 se ha quedado con la tarea 4 y esta desaparece de las disponibles


****************************************************************************************************************************************
****************************************************************************************************************************************
                                        AHORA ASIGNANDO TRABAJADORES A LAS TAREAS CONTANDO CON LA SUMA
****************************************************************************************************************************************
****************************************************************************************************************************************
****************************************************************************************************************************************

----------------------------------------
|
v
6 3 7 5  <-
1 1 7 9
4 1 2 5
6 1 3 8

Asignamos el trabajador 1 a la tarea 1
----------------------------------------
|
v
6 3 7 5
1 1 7 9  <-
4 1 2 5
6 1 3 8

Asignamos el trabajador 2 a la tarea 1
----------------------------------------
|
v
6 3 7 5
1 1 7 9
4 1 2 5  <-
6 1 3 8

----------------------------------------
|
v
6 3 7 5
1 1 7 9
4 1 2 5
6 1 3 8  <-
----------------------------------------
Finalmente el trabajador 2 se ha quedado con la tarea 1 y esta desaparece de las disponibles

----------------------------------------
  |
  v
  3 7 5  <-
1
  1 2 5
  1 3 8

Asignamos el trabajador 1 a la tarea 2

----------------------------------------
  |
  v
  3 7 5
1
  1 2 5  <-
  1 3 8

Asignamos el trabajador 3 a la tarea 2
----------------------------------------
  |
  v
  3 7 5
1
  1 2 5
  1 3 8  <-
----------------------------------------
```

```
-----------------------------------------
Finalmente el trabajador 4 se ha quedado con la tarea 2 y esta desaparece de las disponibles

-----------------------------------------
    |
    v
    7 5  <-
1
    2 5
  1

Asignamos el trabajador 1 a la tarea 3

-----------------------------------------
    |
    v
    7 5
1
    2 5  <-
  1

Asignamos el trabajador 3 a la tarea 3
-----------------------------------------
Finalmente el trabajador 3 se ha quedado con la tarea 3 y esta desaparece de las disponibles

-----------------------------------------
      |
      v
      5  <-
1
    2
  1

Asignamos el trabajador 1 a la tarea 4


-----------------------------------------
Finalmente el trabajador 1 se ha quedado con la tarea 4 y esta desaparece de las disponibles
```

```
Tareas que hace cada trabajador:
El trabajador 1 hace la tarea 2 que tiene un coste de: 3
El trabajador 2 hace la tarea 1 que tiene un coste de: 1
El trabajador 3 hace la tarea 3 que tiene un coste de: 2
El trabajador 4 hace la tarea 4 que tiene un coste de: 8
El coste total asignando tareas es: 14

Trabajador para cada tarea:
La tarea 1 es hecha por el trabajador 2 que tiene un coste de: 1
La tarea 2 es hecha por el trabajador 3 que tiene un coste de: 1
La tarea 3 es hecha por el trabajador 4 que tiene un coste de: 3
La tarea 4 es hecha por el trabajador 1 que tiene un coste de: 5
El coste total asignando trabajadores es: 10

Tareas que hace cada trabajador teniendo en cuenta el coste del resto:
El trabajador 1 hace la tarea 2 que tiene un coste de: 3
El trabajador 2 hace la tarea 1 que tiene un coste de: 1
El trabajador 3 hace la tarea 3 que tiene un coste de: 2
El trabajador 4 hace la tarea 4 que tiene un coste de: 8
El coste total asignando tareas es: 14

Trabajador para cada tarea teniendo en cuenta el coste del resto:
La tarea 1 es hecha por el trabajador 2 que tiene un coste de: 1
La tarea 2 es hecha por el trabajador 4 que tiene un coste de: 1
La tarea 3 es hecha por el trabajador 3 que tiene un coste de: 2
La tarea 4 es hecha por el trabajador 1 que tiene un coste de: 5
El coste total asignando trabajadores es: 9

El coste mas bajo se ha obtenido tras asignar de la siguiente forma:
    Asignando trabajadores a tareas teniendo en cuenta el coste de esos trabajadores para otras tareas, lo cual da un coste de 9 y un reparto (trabajador-tarea):
2 - 1 / 4 - 2 / 3 - 3 / 1 - 4 /



La ejecucion tarda 0.352 segundos
```

Este es uno de esos casos concretos en los que, haber añadido un algoritmo que en caso de empate evalúa más opciones que la de quedarse con el primero que se encuentra, es útil y genera una asignación que cumple con las restricciones y da menos coste que el resto.

Ejemplo 3: Nº de trabajadores = 3 / Nº de tareas = 5. Ejemplo de funcionamiento de los algoritmos extras.

Otro ejemplo que sirve como muestra de que en algunos casos es mejor el algoritmo que tiene en cuenta la suma que el que asigna por orden de encuentro.

```
Tareas que hace cada trabajador:
El trabajador 1 hace la tarea 2 que tiene un coste de: 2
El trabajador 2 hace la tarea 3 que tiene un coste de: 1
El trabajador 3 hace la tarea 5 que tiene un coste de: 6
El trabajador 4 hace la tarea 4 que tiene un coste de: 6
El trabajador 5 hace la tarea 1 que tiene un coste de: 5
El coste total asignando tareas es: 20

Trabajador para cada tarea:
La tarea 1 es hecha por el trabajador 2 que tiene un coste de: 5
La tarea 2 es hecha por el trabajador 1 que tiene un coste de: 2
La tarea 3 es hecha por el trabajador 3 que tiene un coste de: 2
La tarea 4 es hecha por el trabajador 4 que tiene un coste de: 6
La tarea 5 es hecha por el trabajador 5 que tiene un coste de: 7
El coste total asignando trabajadores es: 22

Tareas que hace cada trabajador teniendo en cuenta el coste del resto:
El trabajador 1 hace la tarea 2 que tiene un coste de: 2
El trabajador 2 hace la tarea 4 que tiene un coste de: 1
El trabajador 3 hace la tarea 3 que tiene un coste de: 2
El trabajador 4 hace la tarea 5 que tiene un coste de: 1
El trabajador 5 hace la tarea 1 que tiene un coste de: 5
El coste total asignando tareas es: 11

Trabajador para cada tarea teniendo en cuenta el coste del resto:
La tarea 1 es hecha por el trabajador 5 que tiene un coste de: 5
La tarea 2 es hecha por el trabajador 1 que tiene un coste de: 2
La tarea 3 es hecha por el trabajador 2 que tiene un coste de: 1
La tarea 4 es hecha por el trabajador 4 que tiene un coste de: 6
La tarea 5 es hecha por el trabajador 3 que tiene un coste de: 6
El coste total asignando trabajadores es: 20

El coste mas bajo se ha obtenido tras asignar de la siguiente forma:
    Asignando tareas a trabajadores teniendo en cuenta el coste de esas tareas para otros trabajadores, lo cual da un coste de 11 y un reparto (trabajador-tarea):
1 - 2 / 2 - 4 / 3 - 3 / 4 - 5 / 5 - 1 /
```

Ejemplo 4: Nº de trabajadores = 5 / Nº de tareas = 5. Empate con diferentes asignaciones.

```
Tareas que hace cada trabajador:
El trabajador 1 hace la tarea 5 que tiene un coste de: 2
El trabajador 2 hace la tarea 1 que tiene un coste de: 4
El trabajador 3 hace la tarea 2 que tiene un coste de: 2
El trabajador 4 hace la tarea 4 que tiene un coste de: 5
El trabajador 5 hace la tarea 3 que tiene un coste de: 4
El coste total asignando tareas es: 17

Trabajador para cada tarea:
La tarea 1 es hecha por el trabajador 5 que tiene un coste de: 2
La tarea 2 es hecha por el trabajador 3 que tiene un coste de: 2
La tarea 3 es hecha por el trabajador 2 que tiene un coste de: 6
La tarea 4 es hecha por el trabajador 4 que tiene un coste de: 5
La tarea 5 es hecha por el trabajador 1 que tiene un coste de: 2
El coste total asignando trabajadores es: 17

Tareas que hace cada trabajador teniendo en cuenta el coste del resto:
El trabajador 1 hace la tarea 5 que tiene un coste de: 2
El trabajador 2 hace la tarea 1 que tiene un coste de: 4
El trabajador 3 hace la tarea 2 que tiene un coste de: 2
El trabajador 4 hace la tarea 4 que tiene un coste de: 5
El trabajador 5 hace la tarea 3 que tiene un coste de: 4
El coste total asignando tareas es: 17

Trabajador para cada tarea teniendo en cuenta el coste del resto:
La tarea 1 es hecha por el trabajador 5 que tiene un coste de: 2
La tarea 2 es hecha por el trabajador 3 que tiene un coste de: 2
La tarea 3 es hecha por el trabajador 2 que tiene un coste de: 6
La tarea 4 es hecha por el trabajador 4 que tiene un coste de: 5
La tarea 5 es hecha por el trabajador 1 que tiene un coste de: 2
El coste total asignando trabajadores es: 17

El coste mas bajo se ha obtenido tras asignar de la siguiente forma:
    Asignando tareas a trabajadores, lo cual da un coste de 17 y un reparto (trabajador-tarea):
1 - 5 / 2 - 1 / 3 - 2 / 4 - 4 / 5 - 3 /


    Asignando trabajadores a tareas, lo cual da un coste de 17 y un reparto (trabajador-tarea):
5 - 1 / 3 - 2 / 2 - 3 / 4 - 4 / 1 - 5 /

    Asignando tareas a trabajadores teniendo en cuenta el coste de esas tareas para otros trabajadores, lo cual da un coste de 17 y un reparto (trabajador-tarea):
1 - 5 / 2 - 1 / 3 - 2 / 4 - 4 / 5 - 3 /

    Asignando trabajadores a tareas teniendo en cuenta el coste de esos trabajadores para otras tareas, lo cual da un coste de 17 y un reparto (trabajador-tarea):
5 - 1 / 3 - 2 / 2 - 3 / 4 - 4 / 1 - 5 /
```

En este caso, el coste es el mismo en los 4 algoritmos, pero se llega a él asignando tareas diferentes a trabajadores diferentes. Muestra así, dos alternativas que plantean lo mismo.

También se comprueba que ocurre cuando se mete un número de trabajadores diferentes al número de tareas. En este caso, se añaden tareas extras para que haya una por cada trabajador, y en el caso de que sea al revés, se añadirán trabajadores que puedan realizar esas tareas.