6. Durante la construcción de una casa, se deben recortar seis viguetas de 24 pies cada una a la longitud correcta de 23 pies. La operación de recortar una vigueta implica la siguiente secuencia.

Operación	Tiempo (segundos)
1. Colocar la vigueta en caballetes de aserrar	15
2. Medir la longitud correcta (23 pies)	5
3. Marcar la línea de corte para la sierra circular	5
4. Recortar la vigueta a la longitud correcta	20
5. Apilar las viguetas recortadas en un área designada	20

Intervienen tres personas: Dos deben realizar al mismo tiempo las operaciones 1,2 y 5, y un cortador se ocupa de las operaciones 3 y 4. Hay dos pares de caballetes de aserrar donde se colocan las viguetas sin recortar, y cada par puede manejar tres viguetas. Sugiera un buen plan para cortar las seis viguetas.
Primeramente, daré a las personas Uno y dos las letras A y B para identificarlas y al cortador lo identificamos con la letra C

Parámetros:

1. $n = 6$: numero de viguetas.
2. Tiempos por tarea(segundos)
   - $t_1 = 15$ colocar vigueta
   - $t_2 = 5$ medir longitud
   - $t_3 = 5$ marcar la línea de corte.
   - $t_4 = 20$ recortar la vigueta.
   - $t_5 = 20$ apilar la vigueta.
3. Tiempo de traslado del cortador C: $t_c = 10\ segundos$
4. Asignamos las tareas persona A y B realizan las operaciones 1,2 y 5
5. Persona C realiza las operaciones 3 y 4

Variables de decisión:

$$x_i \rightarrow Tiempo\ en\ el\ que\ comienza\ la\ operación\ i\ en\ la\ secuencia.$$
$$x_i \rightarrow Tiempo\ total\ mínimo\ para\ completar\ el\ proceso.$$

Función objetivo:

$$Minimizar\ T = 6 * (t_1 + t_2 + t_3 + t_4 + t_5) = 6 * 65 = 390\ segundos$$

Restricciones:

Cada Operación debe completarse antes de que la siguiente pueda comenzar:

$$x_1 \leq x_2 \leq x_3 \leq x_4 \leq x_5 \leq T$$

Tiempo acumulado de cada operación para todas las viguetas:

$$x_1 = n * t_1 = 6 * 15 = 90$$
$$x_2 = x_1 + n * t_2 = 90 + 6 * 5 = 120$$
$$x_3 = x_2 + n * t_3 = 120 + 6 * 5 = 150$$
$$x_4 = x_3 + n * t_4 = 150 + 6 * 20 = 270$$
$$x_5 = x_4 + n * t_5 = 270 + 6 * 20 = 390$$

Tiempo de desplazamiento del cortador C:

$$x_3 + t_c \leq x_4$$

Tiempo total maximo del proceso:

$$T \geq x_5$$

```
gloria@MacBook-Air-de-Gloria ~ % /usr/bin/python3 "/Users/gloria/Des
REAS FACU/6to Semestre/Investigación de Operaciones /6to.py"
Tiempo mínimo total: 390.0 segundos
```

7. Se construye una pirámide (bidimensional) en cuatro capas. La capa inferior se compone de los puntos (equidistantes) 1,2,3,4; la siguiente incluye los puntos 5,6 7; la tercera comprende los puntos 8 y 9, y la superior el punto 10. Lo que se quiere es invertir la pirámide (que la capa inferior incluya un punto y la superior cuatro) cambiando de lugar los puntos.

Parámetros:

Estado inicial:

- $Capa\ 1\ (Inferior)$: P_1, P_2, P_3, P_4
- $Capa\ 2$: P_5, P_6, P_7
- $Capa\ 3$: P_8, P_9
- $Capa\ 4\ (superior)$: P_{10}

Estado que buscamos:

- $Capa\ 1\ (inferior)$: P_{10}
- $Capa\ 3$: P_8, P_9
- $Capa\ 2$: P_5, P_6, P_7
- $Capa\ 1\ (superior)$: P_1, P_2, P_3, P_4

Variables de decisión:

Definimos $x_{i,j}$ como una variable binaria que indica si el punto i se mueve a la posición j

$$x_{i,j} = \begin{cases} 1, si\ el\ punto\ i\ se\ mueve\ a\ la\ posición\ j \\ 0, en\ otro\ caso \end{cases}$$

Donde:

$$i \in \{1,2,3,4,5,6,7,8,9,10\}$$
$$j \in \{1,2,3,4,5,6,7,8,9,10\}$$

Función objetivo:

$$\min M = \sum_{i=1}^{10} \sum_{j=1}^{10} x_{i,j}$$

$$\min M = x_{1,10} + x_{2,9} + x_{3,8} + x_{4,7} + x_{5,6} + x_{6,5} + x_{7,4} + x_{8,3} + x_{9,2} + x_{10,1}$$

$$Cada\ x_{i,j}\ representa\ un\ movimiento\ del\ punto\ i\ y\ j.$$

Restricciones:

Cada punto debe asignarse exactamente a una de las posiciones en la nueva pirámide invertida:

$$\sum_{j=1}^{10} x_{i,j} = 1$$

Cada nueva posición en la pirámide invertida debe recibir exactamente un punto

$$\sum_{i=1}^{10} x_{i,j} = 1$$

El movimiento de i a j no es permitido

$$x_{i,j} = 0$$

```
Número mínimo de movimientos necesarios: 10
Pirámide inicial:
1 2 3 4
5 6 7
8 9
10

Pirámide invertida:
10
8 9
5 6 7
1 2 3 4
```

a) Identifique dos soluciones factibles.

Solución 1:

$$P_{10} \rightarrow P_1 \; baja \; directamente$$
$$P_8 \rightarrow P_2 \; baja \; una \; capa$$
$$P_9 \rightarrow P_3 \; baja \; una \; capa$$
$$P_5 \rightarrow P_4 \; baja \; una \; capa$$
$$P_6 \rightarrow P_5 \; baja \; una \; capa$$
$$P_7 \rightarrow P_6 \; baja \; una \; capa$$
$$P_4 \rightarrow P_7 \; baja \; una \; capa$$
$$P_3 \rightarrow P_8 \; baja \; una \; capa$$
$$P_2 \rightarrow P_9 \; baja \; una \; capa$$
$$P_1 \rightarrow P_{10} \; sube \; hasta \; la \; sima$$

Total: 10 movimientos.

Solución 2:

Algunos puntos hacen pasos intermedios antes de su posición final, agregando 2-4 movimientos extra.

- P_{10} se mueve temporalmente fuera.
- P_1,P_2,P_3,P4P1,P_2,P_3,P_4 se recolocan antes de subir.
- P_8,P_9P_8,P_9 bajan primero.
- P_5,P_6,P_7 ,P_5,P_6,P_7 se acomodan.
- P_{10} regresa a su lugar.

Total: 12-14 movimientos.

b) Determine el número mínimo de movimientos necesarios para invertir la pirámide.

La solucion mas factible es la primera, donde cada punto se mueve una vez directamente a su nueva posición. Por lo tanto el número minimo de movimientos es 10.

8. Cuenta con cuatro cadenas y cada una consta de tres eslabones sólidos. Tiene que hacer un brazalete conectado las cuatro cadenas; romper un eslabón cuesta 2 centavos, y volverlo a soldar 3 centavos.

Parámetros:

- Numero de cadenas 4 (cada una con 3 eslabones).
- Costo de romper un eslabón (2 centavos)
- Costo de soldar un eslabón (3 centavos)

Conectar las 4 cadenas formando un brazalete continuo.

Variables de decisión:

$$x_i: número\ de\ eslabones\ rotos\ en\ la\ estrategia\ i.$$
$$y_i: número\ de\ eslabones\ soldados\ en\ la\ estrategia\ i.$$

Función Objetivo:

$$min\ C = \sum_i (2x_i + 3y_i)$$

Sustituyendo

$$C = 2(3) + 3(3) = 6 + 9 = 15\ centavos$$

Donde x_i representa los eslabones rotos.
y_i representa los eslabones soldados

Restricciones:

- Cada eslabón roto debe ser soldado

  $$x = y$$

  Sustituyendo

  $$3 = 3$$
- Se necesitan exactamente 3 eslabones rotos para conectar las 4 cadenas.

  $$x = 3$$
- Se deben soldar exactamente 3 eslabones

  $$y = 3$$
- No se deben romper más de los necesarios

  $$x \leq 3$$
  $$y \leq 3$$

Modelo

$$min\ C = \sum_i (2x_i + 3y_i)$$

s.a

$$x = y$$
$$x \leq 3$$
$$y \leq 3$$

```
gloria@MacBook-Air-de-Gloria ~ % /usr/bin/python3 "/Users/gloria/Desktop/
REAS FACU/6to Semestre/Investigación de Operaciones /8vo.py"
Número óptimo de eslabones rotos: 3.0
Número óptimo de eslabones soldados: 3.0
Costo mínimo total: 15.0 centavos
```

a) Identifique dos soluciones factibles y evalúelaas.

Solución 1: Romper un eslabon de cada una de las tres cadenas.

1. Se rompe 1 eslabon en 3 de las 4 cadenas.
2. Se usa cadena eslabon roto para conectar la cadena siguiente.
3. Se sueldan los 3 eslabones rotos.

Calculamos el costo.

1. Romper 3 eslabones: $3 * 2 = 6\ centavos$.
2. Soldar 3 eslabones: $3 * 3 = 9\ centavos$.
3. Costo total: $6 + 9 = 15\ centavos$.

Solución 2: romperse y usar los eslabones de una sola cadena.

1. Elegimos una cadena y se rompen sus 3 eslabones.
2. Utilizamos esos 3 eslabones para conectar las otras cadenas.
3. Se sueldan los 3 eslabones.

Calculamos el costo.

1. Romper 3 eslabones: $3 * 2 = 6\ centavos$.
2. Soldar 3 eslabones: $3 * 3 = 9\ centavos$.
3. Costo total: $6 + 9 = 15\ centavos$.

b) Determine el costo mínimo para hacer el brazalete.
En ambos casos rompemos y soldamos 3 eslabones, lo que nos da un costo minimo de 15 centavos.

9. Los cuadros de una tabla rectangular de 11 filas y 9 columnas están numerados en secuencia del 1 al 99 con una recompensa monetaria *oculta* de entre 0 y 20 dólares, asignada a cada cuadro. El juego consiste en que un jugador elige un cuadrado seleccionando cualquier número de dos dígitos y luego restando al número seleccionado la suma de sus dos dígitos. El jugador recibe entonces la recompensa asignada al cuadro seleccionado. Sin importar cuántas veces se repita el juego, ¿qué valores monetarios deben asignarse a los 99 cuadros para minimizar la recompensa de los jugadores? Para hacer el juego interesante, asignar $0 a *todos* los cuadros no es una opción.

Parametros :

- N= 99: Número total de cuadrados en la tabla.
- R_i: $Recompensa\ asignada\ al\ cuadrado\ i\ \in \{1,2,\dots,99\}$
- $S(i)$: $suma\ de\ los\ digitos\ del\ mismo\ i, es\ decir, si\ i = 45, entonces\ S(45) = 4 + 5 = 9$
- $r_i\ \in [0,20]$: $Valor\ monetario\ asignado\ a\ cada\ cuadro\ i, donde\ r_i, esta\ entre\ 0\ y\ 20.$

Variables de desición:

$$r_i: valor\ monetario\ asignado\ al\ cuadrado\ i\ \in \{1,2,\dots,99\}$$

Función Objetivo:

$$Minimizar\ Z = \sum_{i=1}^{99} r_i$$

Restricciones:

1. El valor monetario de cada casilla debe estar entre $0 y $20:
$$0 \leq r_i \leq 20,$$
2. No se puede asignar $0 a todas las casillas:
$$\sum_{i=i}^{99} r_i > 0$$