Ejercicio 5

David Corzo

2020 October 27

1. Ejercicio #1

```python
class trabajos:
    def __init__(self, trabajo, plazo, ganancia):
        self.trabajo = trabajo
        self.plazo = plazo
        self.ganancia = ganancia
    def __str__(self):
        return f"trabajo: {self.trabajo}, plazo: {self.plazo}, ganancia: {self.ganancia}"

def swap(a,b):
    a,b = b,a

def work_allocator(w, plazos):
    """
        w: is an array sorted from high to low according to profit.
        plazos: how many time slots are posible to have.
    """
    time_slots = [None for x in range(plazos)]
    trabajo_allocado = ["" for x in range(plazos)]
    counter = 0
    for i in w:
        for ii in range(min( (plazos-1), i.plazo ), -1, -1):
            if (trabajo_allocado[ii] == ""):
                time_slots[ii] = True
                trabajo_allocado[ii] = w[counter]
                break
        counter += 1

    # print(trabajo_allocado)
    counter = 0
    while (counter < len(trabajo_allocado)-1):
        if (counter <= trabajo_allocado[counter].plazo):
            if (counter <= trabajo_allocado[counter + 1].plazo):
                swap(trabajo_allocado[counter], trabajo_allocado[counter + 1])
        elif (trabajo_allocado[counter].plazo == trabajo_allocado[counter + 1].plazo):
            if (trabajo_allocado[counter].ganancia < trabajo_allocado[counter + 1].ganancia):
                swap(trabajo_allocado[counter], trabajo_allocado[counter + 1])
        counter += 1

```

```
    for i in trabajo_allocado:
        print(i)
    print(f"Maximized profit is: {sum([i.ganancia for i in trabajo_allocado])}")

def main():
    w = []
    for trabajo, plazo, ganancia in zip(
            ["T1", "T2", "T3", "T4", "T5", "T6"],
            [5, 3, 3, 2, 4, 2],
            [200, 180, 190, 300, 120, 100]):
        t = trabajos(trabajo, plazo, ganancia)
        w.append(t)
    w = sorted(w, reverse=True, key=lambda o: o.ganancia)
    work_allocator(w, 5)



if __name__ == "__main__":
    main()

# output:
# trabajo: T5, plazo: 4, ganancia: 120
# trabajo: T2, plazo: 3, ganancia: 180
# trabajo: T4, plazo: 2, ganancia: 300
# trabajo: T3, plazo: 3, ganancia: 190
# trabajo: T1, plazo: 5, ganancia: 200
# Maximized profit is: 990
```

(a) Si se llegan a completar todos los tiempos.

(b) La ganancia óptima resulta ser: 990.

(c) Complejidad: la complejidad es n^2.

2. Ejercicio #2

```
class trabajos:
    def __init__(self, name:str, datos:int):
        self.name = name
        self.datos = datos

def main(arr):
    sorted_arr = sorted([x.datos for x in arr])
    print(sorted_arr)
    a = []
    s = sorted_arr[0] + sorted_arr[1]
    for i in sorted_arr[2:]:
        a.append(s)
        s += i
    a.append(s)

    print(f"Result is: {sum(a)}")

```

```python

if __name__ == "__main__":
    arr = list()
    for name,datos in zip(
        ["a","b","c","d","e","f"],
        [40,10,20,15,25,30]):
        l = trabajos(name, datos)
        arr.append(l)
    main(arr)

# output:
# [10, 15, 20, 25, 30, 40]
# Result is: 380
```