

UNIVERSIDAD DE INGENIERÍA
Y TECNOLOGÍA - UTEC

Tarea 4

Fecha de inicio: 19 de Noviembre, 2021
Fecha de entrega: 12 de Diciembre, 2021

Curso: Programación 1 (1100) – Entrega mediante: Gradescope

Indicaciones generales

1. Recuerda que la tarea es **individual**. Los casos de copia/plagio serán sancionados con nota cero (0) en la asignatura.
2. (a) La fecha límite de entrega es el **domingo 12 de Diciembre a las 23:59 hrs.**
(b) Es altamente recomendable no esperar hasta la última hora.
(c) *Gradescope* desactivará automáticamente los envíos pasada dicha hora límite.
(d) **No se aceptarán entregas atrasadas ni entregadas por otros medios.**
3. Revisa bien lo que entregas, aunque en esta oportunidad podrás entregar ilimitadas veces la tarea, la última enviada será la evaluada.
4. Recuerda que *Gradescope* corrige automáticamente tu entrega. Dicha plataforma mostrará si has realizado correctamente las pruebas y mostrará algunos mensajes en color verde. Puedes ver un ejemplo de este caso en el anexo 1.
5. Es posible que hayas subido tu entrega pero hayas modificado algo que no se debió en el template. En ese escenario, *Gradescope* te mostrará algunos mensajes de error. Puedes ver un ejemplo de esto en el anexo 2.

Gradescope

1. Nosotros les proporcionaremos un código base de donde deberán partir para completar dicho ejercicio. Este archivo es llamado `solution.py` y lo encontrarán en la indicación de la tarea en CANVAS.
2. Al finalizar, **solo** subir el archivo `solution.py` (NO cambiar el nombre del archivo y NO comprimirlo).

3. Cada pregunta tiene diversos casos de prueba. Para obtener la nota completa en una pregunta, el algoritmo debe obtener la respuesta correcta en dichos casos de prueba.
4. Si un caso de prueba falla, visualizarán un mensaje de error con sugerencias. **Lee el error**, revisa el código e inténtalo de nuevo.
5. Los input de los casos de prueba son confidenciales.

Indicaciones específicas

1. En el anexo 3, se puede ver la plantilla de código que deberán de seguir para resolver los ejercicios.
2. Ustedes deben escribir dentro de la sección y a la misma altura de donde esta escrito "*Código comienza aquí*". Además, no deben modificar nada debajo de "*Código acaba aquí*". Recuerden tener cuidado con las indentaciones.
3. Los input del ejercicio se encuentran en la plantilla. Recuerden usar estas variables para resolver el ejercicio.
4. La respuesta del ejercicio debe ser retornada según especifique la plantilla otorgada.
5. Al momento de retornar el resultado que pide el ejercicio utilizar la estructura solicitada. En este tarea se requiere trabajar con la estructura de diccionarios para retornar el resultado.
6. No se permite el uso de *built-in functions* que realicen el trabajo por ustedes como las siguientes: `sort`, `sorted`, `max` y `min`.
7. En la plantilla se utiliza la siguiente función: `sys.setrecursionlimit()`. Se conoce que algunos equipos de Windows presentan un error con esta función. Por lo tanto, se les recomienda usar Colab o Replit en ese caso.
8. Las respuestas tendrán el máximo puntaje si y solo si cumplen con la complejidad indicada.
9. En canvas se encuentra el archivo *pokemon.csv* para que puedan realizar sus pruebas.

Laboratorio del Profesor Oak

Te encuentras trabajando como practicante en el laboratorio del Profesor Oak y te ha asignado la tarea de analizar el inventario de Pokémones. Él tiene un archivo que contiene los siguientes datos:

- Name: nombre del Pokémon
- id: número de identificación correspondiente al Pokémon
- sp_attack: puntos de ataque del Pokémon
- sp_defense: puntos de defensa del Pokémon
- speed: puntos de velocidad del Pokémon
- ability: lista de habilidades del Pokémon

Un ejemplo del contenido del archivo y el orden de las columnas se puede observar en el cuadro 1.

name	id	sp_attack	sp_defense	speed	ability
Bulbasaur1	1	65	65	45	Overgrow
Ivysaur1	2	80	80	60	Overgrow
Venusaur1	3	122	120	80	Overgrow
Charmander1	4	60	50	65	Blaze
Charmeleon1	5	80	65	80	Blaze
Charizard1	6	159	115	100	Blaze
...
Baltoy32	19997	40	70	55	Levitate
Claydol32	19998	70	120	75	Levitate
Lileep32	19999	61	87	23	Suction Cups
Cradily32	20000	81	107	43	Suction Cups

Table 1: Tabla *pokemon.csv*

Para trabajar esta información el profesor Oak te pide pasar los datos de los Pokémones a un diccionario que contiene otro diccionario con la siguiente estructura:

- Clave: ID del Pokémon.
- Valor: Diccionario con la siguiente estructura:

- Clave: "Nombre"; Valor: Nombre del Pokémon.
- Clave: "puntos_ataque"; Valor: Sp_attack del Pokémon.
- Clave: "puntos_defensa"; Valor: Sp_defense del Pokémon.
- Clave: "puntos_velocidad"; Valor: Speed del Pokémon.
- Clave: "habilidad"; Valor: Ability del Pokémon

Los ejemplos de los diccionarios a utilizar se muestra en Listing 1.

```
1 {  
2 1: {"nombre": "Bulbasaur1", "puntos_ataque": 65, "puntos_defensa": 65, "  
   puntos_velocidad": 45, "habilidad": "Overgrow"},  
3 2: {"nombre": "Ivysaur1", "puntos_ataque": 80, "puntos_defensa": 80, "  
   puntos_velocidad": 60, "habilidad": "Overgrow"},  
4 ...,  
5 19999: {"nombre": "Lileep32", "puntos_ataque": 61, "puntos_defensa": 87,  
   "puntos_velocidad": 23, "habilidad": "Suction Cups"},  
6 20000: {"nombre": "Cradily32", "puntos_ataque": 81, "puntos_defensa":  
   107, "puntos_velocidad": 43, "habilidad": "Suction Cups"}  
7 }
```

Listing 1: Ejemplo de diccionarios

IMPORTANTE: El ID del Pokémon, los puntos de ataque, puntos de defensa y puntos de velocidad deben ser un entero. El nombre y habilidad deben estar en formato string.

Se recomienda leer el archivo `pokemon.csv` e iterar línea a línea para llenar el diccionario de manera eficiente. Para obtener los valores de la línea separada por punto y coma (;), se recomienda el uso de la función `split()`. Un ejemplo se muestra en Listing 2.

```
1 linea = "Bulbasaur1;1;65;65;45;Overgrow\n"  
2 datos_pokemon = linea.split(";")  
3 print(datos_pokemon)
```

Listing 2: Ejemplo de leer líneas

TIP: Tener en cuenta que al final de la línea existe un carácter no visible de quiebre de línea ("`\n`"), el valor de habilidad no debe contener el carácter mencionado.

Problema 1 - (3 pts)

El profesor Oak le solicita que cree el diccionario a partir del archivo *pokemon.csv*. Recuerde que el diccionario debe tener la siguiente estructura:

- Clave: ID del Pokémon.
- Valor: Diccionario con la siguiente estructura:
 - Clave: "Nombre"; Valor: Nombre del Pokémon.
 - Clave: "puntos_ataque"; Valor: Sp_attack del Pokémon.
 - Clave: "puntos_defensa"; Valor: Sp_defense del Pokémon.
 - Clave: "puntos_velocidad"; Valor: Speed del Pokémon.
 - Clave: "habilidad"; Valor: Ability del Pokémon

Ejemplo 1.

Input :

```
1 ruta = "pokemon.csv"
```

name	id	sp_attack	sp_defense	speed	abilities
Bulbasaur1	1	65	65	45	Overgrow
Ivysaur1	2	80	80	60	Overgrow
Venusaur1	3	122	120	80	Overgrow
Charmander1	4	60	50	65	Blaze
Charmeleon1	5	80	65	80	Blaze
Charizard1	6	159	115	100	Blaze
Squirtle1	7	50	64	43	Torrent
Wartortle1	8	65	80	58	Torrent
Blastoise1	9	135	115	78	Torrent
Caterpie1	10	20	20	45	Shield Dust

Output :

```
1 {
2 1: {'nombre': 'Bulbasaur1', 'puntos_ataque': 65, 'puntos_defensa': 65, '
   puntos_velocidad': 45, 'habilidad': 'Overgrow'},
3 2: {'nombre': 'Ivysaur1', 'puntos_ataque': 80, 'puntos_defensa': 80, '
   puntos_velocidad': 60, 'habilidad': 'Overgrow'},
```

```

4 3: {'nombre': 'Venusaur1', 'puntos_ataque': 122, 'puntos_defensa': 120,
    'puntos_velocidad': 80, 'habilidad': 'Overgrow'},
5 4: {'nombre': 'Charmander1', 'puntos_ataque': 60, 'puntos_defensa': 50,
    'puntos_velocidad': 65, 'habilidad': 'Blaze'},
6 5: {'nombre': 'Charmeleon1', 'puntos_ataque': 80, 'puntos_defensa': 65,
    'puntos_velocidad': 80, 'habilidad': 'Blaze'},
7 6: {'nombre': 'Charizard1', 'puntos_ataque': 159, 'puntos_defensa': 115,
    'puntos_velocidad': 100, 'habilidad': 'Blaze'},
8 7: {'nombre': 'Squirtle1', 'puntos_ataque': 50, 'puntos_defensa': 64, '
    puntos_velocidad': 43, 'habilidad': 'Torrent'},
9 8: {'nombre': 'Wartortle1', 'puntos_ataque': 65, 'puntos_defensa': 80, '
    puntos_velocidad': 58, 'habilidad': 'Torrent'},
10 9: {'nombre': 'Blastoise1', 'puntos_ataque': 135, 'puntos_defensa': 115,
    'puntos_velocidad': 78, 'habilidad': 'Torrent'},
11 10: {'nombre': 'Caterpie1', 'puntos_ataque': 20, 'puntos_defensa': 20, '
    puntos_velocidad': 45, 'habilidad': 'Shield Dust'}
12 }

```

Explicación Se crea el diccionario desde el archivo *pokemon.csv* según la estructura especificada.

Problema 2 - (3 pts)

El profesor Oak necesita consultar frecuentemente datos de los Pokémon que tiene en su laboratorio para intercambiar con otros investigadores. Por ese motivo, él le solicita que genere una función que a partir del ID del Pokémon y un valor de búsqueda devuelva cómo resultado el dato para el Pokémon.

Ejemplo 1.

Input :

```

1 id = 8
2 valor = "habilidad"
3 Pokemones = {
4 1: {'nombre': 'Bulbasaur1', 'puntos_ataque': 65, 'puntos_defensa': 65, '
    puntos_velocidad': 45, 'habilidad': 'Overgrow'},
5 2: {'nombre': 'Ivysaur1', 'puntos_ataque': 80, 'puntos_defensa': 80, '
    puntos_velocidad': 60, 'habilidad': 'Overgrow'},
6 3: {'nombre': 'Venusaur1', 'puntos_ataque': 122, 'puntos_defensa': 120,
    'puntos_velocidad': 80, 'habilidad': 'Overgrow'},
7 4: {'nombre': 'Charmander1', 'puntos_ataque': 60, 'puntos_defensa': 50,
    'puntos_velocidad': 65, 'habilidad': 'Blaze'},
8 5: {'nombre': 'Charmeleon1', 'puntos_ataque': 80, 'puntos_defensa': 65,
    'puntos_velocidad': 80, 'habilidad': 'Blaze'},

```

```

9 6: {'nombre': 'Charizard1', 'puntos_ataque': 159, 'puntos_defensa': 115,
    'puntos_velocidad': 100, 'habilidad': 'Blaze'},
10 7: {'nombre': 'Squirtle1', 'puntos_ataque': 50, 'puntos_defensa': 64, '
    puntos_velocidad': 43, 'habilidad': 'Torrent'},
11 8: {'nombre': 'Wartortle1', 'puntos_ataque': 65, 'puntos_defensa': 80, '
    puntos_velocidad': 58, 'habilidad': 'Torrent'},
12 9: {'nombre': 'Blastoise1', 'puntos_ataque': 135, 'puntos_defensa': 115,
    'puntos_velocidad': 78, 'habilidad': 'Torrent'},
13 10: {'nombre': 'Caterpie1', 'puntos_ataque': 20, 'puntos_defensa': 20, '
    puntos_velocidad': 45, 'habilidad': 'Shield Dust'}
14 }

```

Output :

```
1 Torrent
```

Explicación Se busca al Pokémon por su ID y se retorna el valor de la habilidad, en este caso el Pokémon 8 tiene la habilidad Torrent.

Ejemplo 2.

Input :

```

1 id = 88
2 valor = "puntos_velocidad"
3 Pokemones = {
4 1: {'nombre': 'Bulbasaur1', 'puntos_ataque': 65, 'puntos_defensa': 65, '
    puntos_velocidad': 45, 'habilidad': 'Overgrow'},
5 2: {'nombre': 'Ivysaur1', 'puntos_ataque': 80, 'puntos_defensa': 80, '
    puntos_velocidad': 60, 'habilidad': 'Overgrow'},
6 3: {'nombre': 'Venusaur1', 'puntos_ataque': 122, 'puntos_defensa': 120,
    'puntos_velocidad': 80, 'habilidad': 'Overgrow'},
7 4: {'nombre': 'Charmander1', 'puntos_ataque': 60, 'puntos_defensa': 50,
    'puntos_velocidad': 65, 'habilidad': 'Blaze'},
8 5: {'nombre': 'Charmeleon1', 'puntos_ataque': 80, 'puntos_defensa': 65,
    'puntos_velocidad': 80, 'habilidad': 'Blaze'},
9 6: {'nombre': 'Charizard1', 'puntos_ataque': 159, 'puntos_defensa': 115,
    'puntos_velocidad': 100, 'habilidad': 'Blaze'},
10 7: {'nombre': 'Squirtle1', 'puntos_ataque': 50, 'puntos_defensa': 64, '
    puntos_velocidad': 43, 'habilidad': 'Torrent'},
11 8: {'nombre': 'Wartortle1', 'puntos_ataque': 65, 'puntos_defensa': 80, '
    puntos_velocidad': 58, 'habilidad': 'Torrent'},
12 9: {'nombre': 'Blastoise1', 'puntos_ataque': 135, 'puntos_defensa': 115,
    'puntos_velocidad': 78, 'habilidad': 'Torrent'},
13 10: {'nombre': 'Caterpie1', 'puntos_ataque': 20, 'puntos_defensa': 20, '
    puntos_velocidad': 45, 'habilidad': 'Shield Dust'}
14 }

```

Output :

```
1 Pokémon no encontrado
```

Explicación Se busca al Pokémon por su ID, en este caso no se encuentra el ID y se retorna "Pokémon no encontrado".

Ejemplo 3.**Input :**

```
1 id = 3
2 valor = "puntos_ataque"
3 Pokemones = {
4 1: {'nombre': 'Bulbasaur1', 'puntos_ataque': 65, 'puntos_defensa': 65, '
    puntos_velocidad': 45, 'habilidad': 'Overgrow'},
5 2: {'nombre': 'Ivysaur1', 'puntos_ataque': 80, 'puntos_defensa': 80, '
    puntos_velocidad': 60, 'habilidad': 'Overgrow'},
6 3: {'nombre': 'Venusaur1', 'puntos_ataque': 122, 'puntos_defensa': 120,
    'puntos_velocidad': 80, 'habilidad': 'Overgrow'},
7 4: {'nombre': 'Charmander1', 'puntos_ataque': 60, 'puntos_defensa': 50,
    'puntos_velocidad': 65, 'habilidad': 'Blaze'},
8 5: {'nombre': 'Charmeleon1', 'puntos_ataque': 80, 'puntos_defensa': 65,
    'puntos_velocidad': 80, 'habilidad': 'Blaze'},
9 6: {'nombre': 'Charizard1', 'puntos_ataque': 159, 'puntos_defensa': 115,
    'puntos_velocidad': 100, 'habilidad': 'Blaze'},
10 7: {'nombre': 'Squirtle1', 'puntos_ataque': 50, 'puntos_defensa': 64, '
    puntos_velocidad': 43, 'habilidad': 'Torrent'},
11 8: {'nombre': 'Wartortle1', 'puntos_ataque': 65, 'puntos_defensa': 80, '
    puntos_velocidad': 58, 'habilidad': 'Torrent'},
12 9: {'nombre': 'Blastoise1', 'puntos_ataque': 135, 'puntos_defensa': 115,
    'puntos_velocidad': 78, 'habilidad': 'Torrent'},
13 10: {'nombre': 'Caterpie1', 'puntos_ataque': 20, 'puntos_defensa': 20, '
    puntos_velocidad': 45, 'habilidad': 'Shield Dust'}
14 }
```

Output :

```
1 122
```

Explicación Se busca al Pokémon por su ID y se retorna el valor de puntos_ataque, en este caso el Pokémon 3 tiene 122 puntos de ataque.

Problema 3 - (4 pts)

El profesor Oak requiere conocer el Pokémon más rápido que tiene en su laboratorio para enviarlo a una competencia en el torneo Pokémon de la ciudad Johto. Él le solicita que cree un algoritmo de búsqueda con **Complejidad** $\mathcal{O}(n)$, que retorne una tupla con el nombre del Pokémon y el puntaje de velocidad.

Ejemplo 1.**Input :**

```
1 Pokemones = {
2 1: {'nombre': 'Bulbasaur1', 'puntos_ataque': 65, 'puntos_defensa': 65, '
   puntos_velocidad': 45, 'habilidad': 'Overgrow'},
3 2: {'nombre': 'Ivysaur1', 'puntos_ataque': 80, 'puntos_defensa': 80, '
   puntos_velocidad': 60, 'habilidad': 'Overgrow'},
4 3: {'nombre': 'Venusaur1', 'puntos_ataque': 122, 'puntos_defensa': 120,
   'puntos_velocidad': 80, 'habilidad': 'Overgrow'},
5 4: {'nombre': 'Charmander1', 'puntos_ataque': 60, 'puntos_defensa': 50,
   'puntos_velocidad': 65, 'habilidad': 'Blaze'},
6 5: {'nombre': 'Charmeleon1', 'puntos_ataque': 80, 'puntos_defensa': 65,
   'puntos_velocidad': 80, 'habilidad': 'Blaze'},
7 6: {'nombre': 'Charizard1', 'puntos_ataque': 159, 'puntos_defensa': 115,
   'puntos_velocidad': 100, 'habilidad': 'Blaze'},
8 7: {'nombre': 'Squirtle1', 'puntos_ataque': 50, 'puntos_defensa': 64, '
   puntos_velocidad': 43, 'habilidad': 'Torrent'},
9 8: {'nombre': 'Wartortle1', 'puntos_ataque': 65, 'puntos_defensa': 80, '
   puntos_velocidad': 58, 'habilidad': 'Torrent'},
10 9: {'nombre': 'Blastoise1', 'puntos_ataque': 135, 'puntos_defensa': 115,
   'puntos_velocidad': 78, 'habilidad': 'Torrent'},
11 10: {'nombre': 'Caterpie1', 'puntos_ataque': 20, 'puntos_defensa': 20, '
   puntos_velocidad': 45, 'habilidad': 'Shield Dust'}
12 }
```

Output :

```
1 ('Charizard1', 100)
```

Explicación La búsqueda en el diccionario de Pokémones permite encontrar que el más rápido tiene 100 puntos_velocidad y el nombre es Charizard1.

Problema 4 - (5 pts)

El profesor Oak necesita ordenar el nombre de los Pokémones de forma ascendente para publicar la lista de en la página web del laboratorio. Él le solicita que cree un algoritmo de ordenamiento con **Complejidad** $\mathcal{O}(n \lg n)$, que retorne una lista de tuplas que contenga el nombre del Pokémon y el ID de Pokémon.

TIP: Para implementar las funciones de ordenamiento usando recursividad se le proporciona dos funciones llamadas `merge_sort` y `merge` que debe implementar en la plantilla `Solution`. En la plantilla mostrada en el anexo 3, existe una sección entre las líneas 4 y 23 donde puede editar sus funciones de ordenamiento.

Ejemplo 1.

Input :

```
1 Pokemones = {
2 1: {'nombre': 'Bulbasaur1', 'puntos_ataque': 65, 'puntos_defensa': 65, '
   puntos_velocidad': 45, 'habilidad': 'Overgrow'},
3 2: {'nombre': 'Ivysaur1', 'puntos_ataque': 80, 'puntos_defensa': 80, '
   puntos_velocidad': 60, 'habilidad': 'Overgrow'},
4 3: {'nombre': 'Venusaur1', 'puntos_ataque': 122, 'puntos_defensa': 120,
   'puntos_velocidad': 80, 'habilidad': 'Overgrow'},
5 4: {'nombre': 'Charmander1', 'puntos_ataque': 60, 'puntos_defensa': 50,
   'puntos_velocidad': 65, 'habilidad': 'Blaze'},
6 5: {'nombre': 'Charmeleon1', 'puntos_ataque': 80, 'puntos_defensa': 65,
   'puntos_velocidad': 80, 'habilidad': 'Blaze'},
7 6: {'nombre': 'Charizard1', 'puntos_ataque': 159, 'puntos_defensa': 115,
   'puntos_velocidad': 100, 'habilidad': 'Blaze'},
8 7: {'nombre': 'Squirtle1', 'puntos_ataque': 50, 'puntos_defensa': 64, '
   puntos_velocidad': 43, 'habilidad': 'Torrent'},
9 8: {'nombre': 'Wartortle1', 'puntos_ataque': 65, 'puntos_defensa': 80, '
   puntos_velocidad': 58, 'habilidad': 'Torrent'},
10 9: {'nombre': 'Blastoise1', 'puntos_ataque': 135, 'puntos_defensa': 115,
   'puntos_velocidad': 78, 'habilidad': 'Torrent'},
11 10: {'nombre': 'Caterpie1', 'puntos_ataque': 20, 'puntos_defensa': 20, '
   puntos_velocidad': 45, 'habilidad': 'Shield Dust'}
12 }
```

Output :

```
1 [('Blastoise1', 9), ('Bulbasaur1', 1), ('Caterpie1', 10), ('Charizard1',
   6), ('Charmander1', 4), ('Charmeleon1', 5), ('Ivysaur1', 2), ('
   Squirtle1', 7), ('Venusaur1', 3), ('Wartortle1', 8)]
```

Explicación Se crea una lista de tuplas con los valores de nombre e ID del Pokémon. Luego se realizó el ordenamiento de la lista de forma eficiente con las funciones recursivas. Como resultado se obtiene el nombre de los Pokémones ordenados ascendentemente.

Problema 5 - (5 pts)

En tres semanas el profesor Oak necesita entregar Pokémones a los nuevos entrenadores que llegan a su laboratorio. En esta ocasión va a realizar un sorteo con los nombres de los Pokémones publicados en la página web del laboratorio. Luego del sorteo necesita conocer la habilidad que el Pokémon tiene para instruir a los entrenadores, por ese motivo le solicita que cree un algoritmo de búsqueda binaria con **Complejidad** $\mathcal{O}(\lg n)$, que reciba como parámetro el nombre del Pokémon a buscar y retorne los datos del Pokémon en el diccionario. Si el nombre del Pokémon no es encontrado debe retornar el valor de "-1" como llave y el texto "No encontrado" como valor.

TIP: Para implementar la búsqueda binaria, tiene que trabajar sobre una lista ordenada. Por ese motivo, uno de los parámetros de entrada es la lista de tuplas con los valores de nombre e ID de Pokémon ordenados ascendentemente por el nombre, similar al resultado del Problema 4.

Ejemplo 1.

Input :

```
1 nombre_a_buscar = "Squirtle1"
2 nombres_ordenados = [('Blastoise1', 9), ('Bulbasaur1', 1), ('Caterpie1',
    10), ('Charizard1', 6), ('Charmander1', 4), ('Charmeleon1', 5), ('
    Ivysaur1', 2), ('Squirtle1', 7), ('Venusaur1', 3), ('Wartortle1', 8)]
3 Pokemones = {
4 1: {'nombre': 'Bulbasaur1', 'puntos_ataque': 65, 'puntos_defensa': 65, '
    puntos_velocidad': 45, 'habilidad': 'Overgrow'},
5 2: {'nombre': 'Ivysaur1', 'puntos_ataque': 80, 'puntos_defensa': 80, '
    puntos_velocidad': 60, 'habilidad': 'Overgrow'},
6 3: {'nombre': 'Venusaur1', 'puntos_ataque': 122, 'puntos_defensa': 120,
    'puntos_velocidad': 80, 'habilidad': 'Overgrow'},
7 4: {'nombre': 'Charmander1', 'puntos_ataque': 60, 'puntos_defensa': 50,
    'puntos_velocidad': 65, 'habilidad': 'Blaze'},
8 5: {'nombre': 'Charmeleon1', 'puntos_ataque': 80, 'puntos_defensa': 65,
    'puntos_velocidad': 80, 'habilidad': 'Blaze'},
9 6: {'nombre': 'Charizard1', 'puntos_ataque': 159, 'puntos_defensa': 115,
    'puntos_velocidad': 100, 'habilidad': 'Blaze'},
10 7: {'nombre': 'Squirtle1', 'puntos_ataque': 50, 'puntos_defensa': 64, '
    puntos_velocidad': 43, 'habilidad': 'Torrent'},
```

```

11 8: {'nombre': 'Wartortle1', 'puntos_ataque': 65, 'puntos_defensa': 80, '
    puntos_velocidad': 58, 'habilidad': 'Torrent'},
12 9: {'nombre': 'Blastoise1', 'puntos_ataque': 135, 'puntos_defensa': 115,
    'puntos_velocidad': 78, 'habilidad': 'Torrent'},
13 10: {'nombre': 'Caterpie1', 'puntos_ataque': 20, 'puntos_defensa': 20, '
    puntos_velocidad': 45, 'habilidad': 'Shield Dust'}
14 }

```

Output :

```

1 {7: {'nombre': 'Squirtle1', 'puntos_ataque': 50, 'puntos_defensa': 64, '
    puntos_velocidad': 43, 'habilidad': 'Torrent'}}

```

Explicación : El nombre del Pokémon a buscar es Squirtle1, para obtener el ID del Pokémon se tiene que realizar la búsqueda binaria en el la lista nombres_ordenados. Luego de encontrarlo, tiene que retornar todos los valores del diccionario para ese Pokémon. En el ejemplo obtenemos:

- Clave: 7.
- Valor: Diccionario con la siguiente estructura:
 - Clave: "Nombre"; Valor: Squirtle1.
 - Clave: "puntos_ataque"; Valor: 50.
 - Clave: "puntos_defensa"; Valor: 64.
 - Clave: "puntos_velocidad"; Valor: 43.
 - Clave: "habilidad"; Valor: Torrent.

Ejemplo 2.

Input :

```

1 nombre_a_buscar = "Squirtle5"
2 Pokemones = {
3 1: {'nombre': 'Bulbasaur1', 'puntos_ataque': 65, 'puntos_defensa': 65, '
    puntos_velocidad': 45, 'habilidad': 'Overgrow'},
4 2: {'nombre': 'Ivysaur1', 'puntos_ataque': 80, 'puntos_defensa': 80, '
    puntos_velocidad': 60, 'habilidad': 'Overgrow'},
5 3: {'nombre': 'Venusaur1', 'puntos_ataque': 122, 'puntos_defensa': 120,
    'puntos_velocidad': 80, 'habilidad': 'Overgrow'},
6 4: {'nombre': 'Charmander1', 'puntos_ataque': 60, 'puntos_defensa': 50,
    'puntos_velocidad': 65, 'habilidad': 'Blaze'},
7 5: {'nombre': 'Charmeleon1', 'puntos_ataque': 80, 'puntos_defensa': 65,
    'puntos_velocidad': 80, 'habilidad': 'Blaze'},

```

```
8 6: {'nombre': 'Charizard1', 'puntos_ataque': 159, 'puntos_defensa': 115,
    'puntos_velocidad': 100, 'habilidad': 'Blaze'},
9 7: {'nombre': 'Squirtle1', 'puntos_ataque': 50, 'puntos_defensa': 64, '
    puntos_velocidad': 43, 'habilidad': 'Torrent'},
10 8: {'nombre': 'Wartortle1', 'puntos_ataque': 65, 'puntos_defensa': 80, '
    puntos_velocidad': 58, 'habilidad': 'Torrent'},
11 9: {'nombre': 'Blastoise1', 'puntos_ataque': 135, 'puntos_defensa': 115,
    'puntos_velocidad': 78, 'habilidad': 'Torrent'},
12 10: {'nombre': 'Caterpie1', 'puntos_ataque': 20, 'puntos_defensa': 20, '
    puntos_velocidad': 45, 'habilidad': 'Shield Dust'}
13 }
```

Output :

```
1 {-1: "No encontrado"}
```

Explicación : El nombre del Pokémon a buscar es Squirtle5, para obtener el ID del Pokémon se tiene que realizar la búsqueda binaria en el la lista nombres_ordenados. Debido a que no se encontró al Pokémon se devuelve -1 como clave y "No encontrado" como valor. .

1. Anexos

The screenshot displays the Gradescope interface for 'Tarea 4'. On the left, a sidebar contains navigation links: 'Configure Autograder', 'Manage Submissions', 'Review Grades', 'Regrade Requests', 'Extensions', 'Statistics', 'Review Similarity', and 'Settings'. The main area, titled 'Autograder Results', shows a list of 10 test cases, all of which passed. Each test case is represented by a light blue box with a green checkmark icon. The test cases are: 'Pregunta 1: creación de diccionario con set de datos 1 (1.5/1.5)', 'Pregunta 5: Búsqueda binaria con nombre de Pokémon, usando set de datos 1 (2.0/2.0)', 'Pregunta 5: Búsqueda binaria con nombre de Pokémon inexistente, usando set de datos 1 (1.0/1.0)', 'Pregunta 5: Búsqueda binaria con nombre de Pokémon, usando set de datos 2 (2.0/2.0)', 'Pregunta 1: creación de diccionario con set de datos 2 (1.5/1.5)', 'Pregunta 2: Buscar dato de Pokémon inexistente con set de datos 1 (1.0/1.0)', 'Pregunta 2: Buscar dato de Pokémon con set de datos 1 (1.0/1.0)', 'Pregunta 2: Buscar dato de Pokémon con set de datos 2 (1.0/1.0)', 'Pregunta 3: Pokémon más rápido con set de datos 1 (2.0/2.0)', 'Pregunta 3: Pokémon más rápido con set de datos 2 (2.0/2.0)', 'Pregunta 4: Diccionario ordenado ascendentemente por nombre de Pokémon con set de datos 1 (2.5/2.5)', and 'Pregunta 4: Diccionario ordenado ascendentemente por nombre de Pokémon con set de datos 2 (2.5/2.5)'. On the right, a summary box shows the student's name 'Jonathan Silva Mercado', the 'AUTOGRADER SCORE' as '20.0 / 20.0', and a list of 'PASSED TESTS' corresponding to the 10 test cases.

Figure 1: Casos de prueba correctos en Gradescope.

The screenshot displays the Gradescope interface for 'Tarea 4', showing a failed submission. The 'Autograder Results' section shows 10 test cases, all of which failed. Each test case is represented by a light blue box with a red 'X' icon. The test cases are: 'Pregunta 1: creación de diccionario con set de datos 1 (0.0/1.5)', 'Pregunta 5: Búsqueda binaria con nombre de Pokémon, usando set de datos 1 (0.0/2.0)', 'Pregunta 5: Búsqueda binaria con nombre de Pokémon inexistente, usando set de datos 1 (0.0/1.0)', 'Pregunta 5: Búsqueda binaria con nombre de Pokémon, usando set de datos 2 (0.0/2.0)', 'Pregunta 1: creación de diccionario con set de datos 2 (0.0/1.5)', 'Pregunta 2: Buscar dato de Pokémon inexistente con set de datos 1 (0.0/1.0)', 'Pregunta 2: Buscar dato de Pokémon con set de datos 1 (0.0/1.0)', 'Pregunta 2: Buscar dato de Pokémon con set de datos 2 (0.0/1.0)', 'Pregunta 3: Pokémon más rápido con set de datos 1 (0.0/2.0)', 'Pregunta 3: Pokémon más rápido con set de datos 2 (0.0/2.0)', 'Pregunta 4: Diccionario ordenado ascendentemente por nombre de Pokémon con set de datos 1 (0.0/2.5)', and 'Pregunta 4: Diccionario ordenado ascendentemente por nombre de Pokémon con set de datos 2 (0.0/2.5)'. On the right, a summary box shows the student's name 'Jonathan Silva Mercado', the 'AUTOGRADER SCORE' as '0.0 / 20.0', and a list of 'FAILED TESTS' corresponding to the 10 test cases. The failure messages for each test case are visible in the 'Test Failed' section of each box.

Figure 2: Entrega incorrecta en Gradescope.

```
1 import sys
2 sys.setrecursionlimit(20000)
3
4 # FUNCIONES RECURSIVAS EMPIEZAN AQUI
5
6 def merge(left, right):
7     merged_list = []
8     # SU SOLUCION EMPIEZA AQUI
9
10
11
12     # SU SOLUCION TERMINA AQUI
13     return merged_list
14
15 def merge_sort(lista):
16     # SU SOLUCION EMPIEZA AQUI
17     left = () # debe implementar el valor correcto de left
18     right = () # debe implementar el valor correcto de right
19
20     # SU SOLUCION TERMINA AQUI
21     return merge(left, right)
22
23 # FUNCIONES RECURSIVAS TERMINAN AQUI
24
25 class Solution:
26
27     # NO MODIFICAR ABAJO DE EST LINEA, ES PARTE DEL AUTOGRADER
28     def sort(self, data=[]):
29         return "clear"
30
31     def sorted(self, data=[]):
32         return "clear"
33     # NO MODIFICAR ARRIBA DE EST LINEA, ES PARTE DEL AUTOGRADER
34
35     # ===== Pregunta 1=====
36
37     def crear_diccionarios(self, ruta="pokemon.csv"):
38         pokemones={}
39         # SU SOLUCION EMPIEZA AQUI
40
41         # SU SOLUCION TERMINA AQUI
42         return pokemones
43
44     # ===== Pregunta 2=====
45     def buscar_dato_pokemon(self, pokemones, id, valor):
46         result = ""
47         # SU SOLUCION EMPIEZA AQUI
```

```
48
49
50     # SU SOLUCION TERMINA AQUI
51     return result
52
53     # ===== Pregunta 3=====
54     def pokemon_rapido(self, pokemones):
55         result = ()
56         # SU SOLUCION EMPIEZA AQUI
57
58         # SU SOLUCION TERMINA AQUI
59         return result
60
61     # ===== Pregunta 4=====
62     def nombre_ascendente(self, pokemones):
63         result = []
64         # SU SOLUCION EMPIEZA AQUI
65
66
67         # SU SOLUCION TERMINA AQUI
68         return result
69
70     # ===== Pregunta 5=====
71     def busqueda_habilidad(self, nombre_a_buscar, nombres_ordenados,
72 pokemones):
73         result = {}
74         # SU SOLUCION EMPIEZA AQUI
75
76
77         # SU SOLUCION TERMINA AQUI
78         return result
```

Listing 3: Template solution.py.