```
import time

def diferenciar_arquivos(dir_arquivo_a, dir_arquivo_b, dir_arquivo_resultado):
    """Este método recebe o caminho do diretório de 2 arquivos .csv, dir_arquivo_a
    e dir_arquivo_b, e o diretório do arquivo de saída dir_resultado. O método
    efetua a leitura dos dois arquivos e compara os conteúdos, retornando no
    arquivo de saída a diferença entre os dados do arquivo_a e o arquivo_b.

    Inputs:         dir_arquivo_a -> local onde está o arquivo a
                    dir_arquivo_a -> local onde está o arquivo b
            dir_arquivo_resultado -> local onde será salvo o arquivo resultado
    Outputs:           um arquivo -> arquivo_resultado.csv
    """

    # Parte 1: Leitura dos dois arquivos e armazenamento dos dados em uma lista

    lista_a = []
    arquivo_a = open(dir_arquivo_a, "r")
    lista_a = arquivo_a.readlines()
    arquivo_a.close()

    lista_b = []
    arquivo_b = open(dir_arquivo_b, "r")
    lista_b = arquivo_b.readlines()
    arquivo_b.close()

    # Parte 2: Converte a lista 2 em um conjunto (set), a fim de usar
    # um loop para checar se os dados da lista_a estão dentro deste set

    lista_b_set = set(lista_b)

    # Usando List Comprehensions vamos criar uma lista com os valoresda lista_a
    # que não estejam contidos no set lista_b_set
    lista_resultado = [x for x in lista_a if x not in lista_b_set]

    # Também poderíamos transformar as duas listas em sets e fazer set(a) - set(b)
    # porém, neste caso iríamos perder a informação da ordem de aparição na lista a

    # Parte 3: Criar o arquivo resultado e adicionar os dados da lista_resultado

    tamanho_lista_resultado = range(len(lista_resultado))

    arquivo_resultado = open(dir_arquivo_resultado, "w+")
    for linha in tamanho_lista_resultado:
        arquivo_resultado.write(lista_resultado[linha])
    arquivo_resultado.close()

""" Vamos testar com 2 arquivos de exemplo:
    arquivo_a.csv tem 1.048.576 ids
    arquivo_b.csv tem 800.000 ids
As ids do arquivo_b são as mesmas 800.000 primeiras ids do arquivo_a.csv.
Dessa forma, o arquivo de saída terá 1.048.576 - 800.000 = 248.576 ids.
Vamos contar o tempo de execução para ter uma ideia da performance do método.
Iremos executar o método 5 vezes em um loop para simular uma quantidade de ids
```

```
    Iremos executar o método 5 vezes em um loop para simular uma quantidade de 1gb
    acima de 5 milhões, conforme enunciado do desafio.
    """

    dir_arquivo_a = "/arquivo_a.csv"
    dir_arquivo_b = "/arquivo_b.csv"
    dir_arquivo_resultado = "/arquivo_resultado.csv"

    start_time = time.time()
    for k in range(5):
        diferenciar_arquivos(dir_arquivo_a, dir_arquivo_b, dir_arquivo_resultado)
    print(f'{time.time() - start_time} segundos de execução para este método.')
```

```
4.574575424194336 segundos de execução para este método.
```

✓ 4s conclusão: 14:37