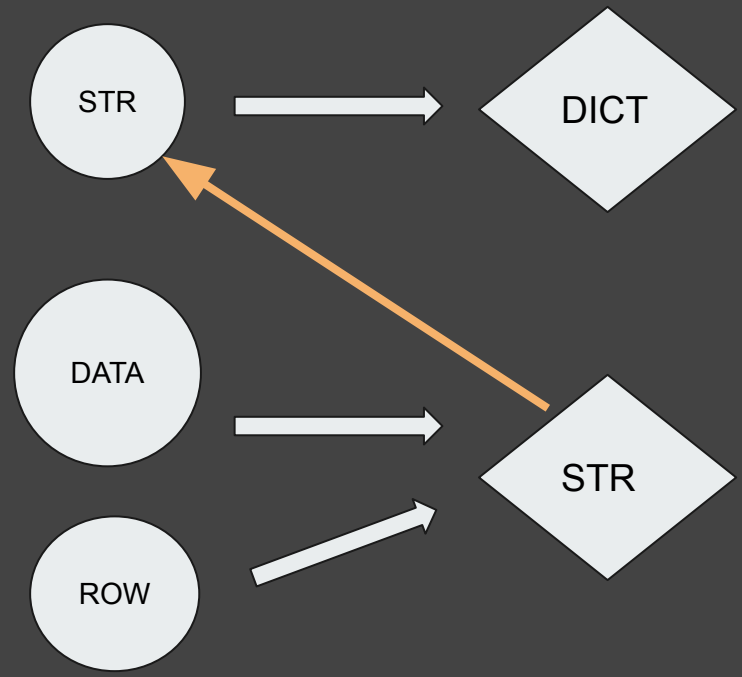
BIG Data Presentation

Prasham Bhuta,
Goda Klaudija Kovalenkinaite

Subtask

```
def count_chars(string_text):
    """
    Counts and returns the characters dict inside a string.

    :param string_text: just a string
    :return: a dict with key as character, and value as count
    """

def character_count(data, row_number):
    """
    Count the characters inside the string. return dict with
following format :
    {character: number of times it appears}

    :param data: nx2, where col1 = row_number, col2 = text
    :param row_number: the row_number for the task
    :return: str
    """
```

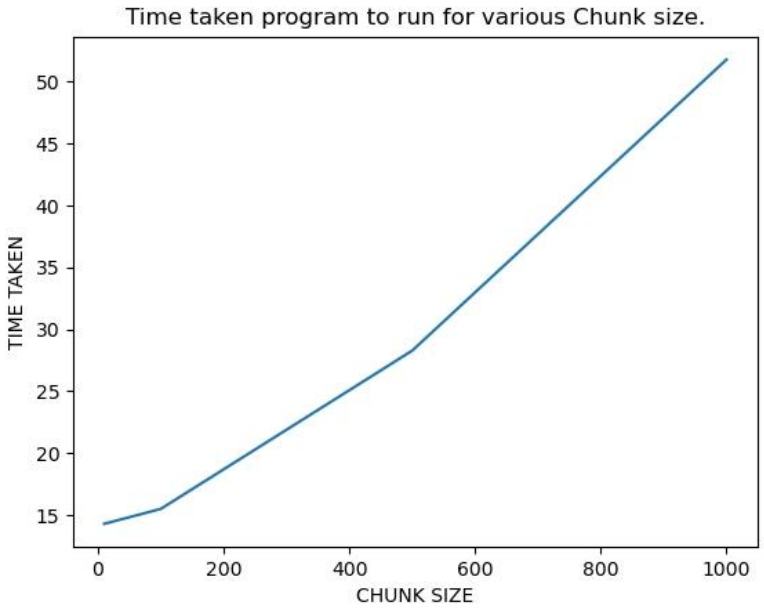


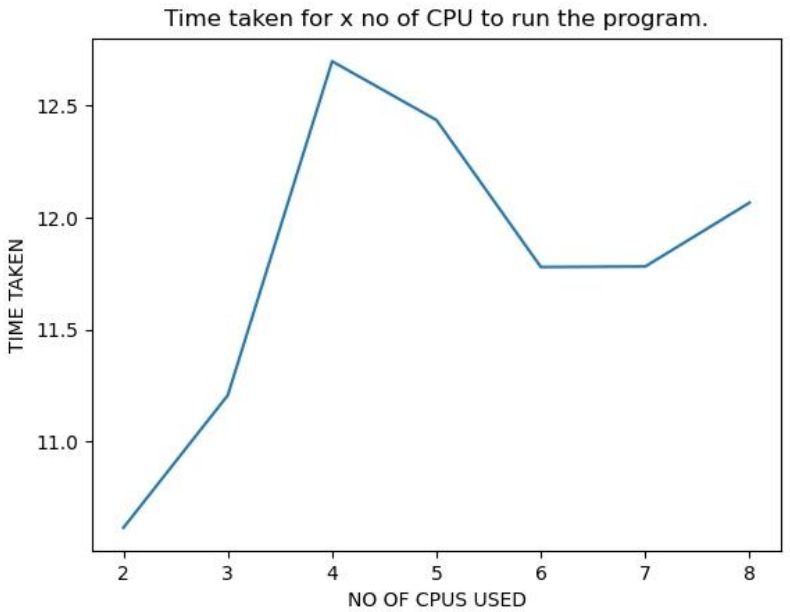
Parallel Computing

```
# Number of maximum cpu_core available
cpus = mp.cpu_count()

# Init Pool of `multiprocessing` library
pool = mp.Pool(cpus)

# Parallelising using Pool.apply()
results = [pool.apply(character_count, args=(chunks, row)) for
row in row_number]
```

Parallel Computing based on chunks, and unique rows.

Complexity plots





Results

Unique Characters:

There are 66 unique characters in the `abstract` column of the dataset.

Top 10 Characters.

‘ ’	n
e	o
t	s
i	r
a	d

Thank you.