

Escolha e baixe um arquivo csv a partir do link e use-o nas questões a seguir. Na resolução desta questão, não enviar os arquivos envolvidos, mas somente os resultados pedidos em arquivo PDF.

Requisitos:

- Gere as somas md5 e sha1 do arquivo encriptado.
- Compacte via linha de comando o arquivo csv original usando compressões zip, gzip, bzip2, rar e 7zip. Depois disso compare os tempos e tamanhos dos arquivos gerados. Dica: para obter o tempo de execução, use o comando time. Exemplo: `time zip iris.zip iris.csv`

Sobre a minha resolução

- Para reproduzir meus testes (caso queira) deixei um script sh onde é baixado um arquivo csv, e nele são realizados:
 - Testes de hashing com md5sum e sha1sum do arquivo data.csv
 - Testes dos tempos de compactação entre cada desses formatos:
 - Zip
 - Gzip
 - Bzip2
 - 7zip
 - Rar
- Ainda quando cada arquivo compactado é criado é medido o tempo e o arquivo compactado é deletado.
- Da mesma maneira o data.csv, caso ele não esteja na sua máquina será baixado e no final do script será excluído.
- Como resultado será gerado dois arquivos com extensão `.txt` com os seguintes resultados:
 - solve-hash.txt: Arquivo contendo as chaves hasheadas que foram geradas em cada um dos scripts (sha1sum e md5sum)
 - solve-compression.txt: Sobre o resultado dos tempos para cada tipo de compactação
- Para executar o scripts rode:

```
bash script.sh
```

Resultados

- Ao abrir o arquivo `solve-hash.txt` você verá os seguintes resultados:

```
--- Hashing data.csv ---
```

```
md5sum: 75c9e7b5dd6548ebf9d879937905931c data.csv
sha1sum: aa0cca517cc4261cb28a8b0829e6e75e8ecd589d data.csv
```

- Ao abrir o arquivo `solve-compression.txt` verá os seguintes resultados:

```
-- Time and size between each type of compression --
```

zip:

```
real    0m13,259s
user    0m13,141s
sys 0m0,100s
-rw-rw-r-- 1 henricker henricker 79M out 10 19:09 data.zip
```

gzip:

```
real    0m11,791s
user    0m11,718s
sys 0m0,052s
-rw-rw-r-- 1 henricker henricker 79M out 10 19:10 data.gz
```

bzip2:

```
real    0m16,132s
user    0m16,001s
sys 0m0,112s
-rw-rw-r-- 1 henricker henricker 57M out 8 22:24 data.csv.bz2
```

7zip:

```
real    1m30,151s
user    3m52,404s
sys 0m1,358s
-rw-rw-r-- 1 henricker henricker 51M out 10 19:12 data.7z
```

rar:

```
real    0m31,126s
user    1m42,275s
```

```
sys 0m0,451s
```

```
-rw-rw-r-- 1 henricker henricker 60M out 10 19:12 data.rar
```

- Com o resultado acima podemos fazer uma tabela para melhor visualização dos dados

type compression	time (s)	size (M)
zip	13.259	79
gzip	11.791	79
bzip2	16.132	57
7zip	90.151	51
rar	31.126	60

- Pela tabela acima é perceptível que o melhor desempenho foi o `bzip2` pois apresentou o menor tamanho comprimido e em um tempo razoável de 16 segundos.
- Acredito que o tempo possa variar de máquina para máquina, mas estes foram meus resultados.