Blai Ras i Arnau Vancells

Compressió LZ-77: Text

Execució

Per a executar el programa se segueix el mateix esquema que en la pràctica anterior.

Per a fer proves només cal fer servir -mode 3, -input "path_fitxer_text", i els valors -mDes i -mEnt que vulguem.

Un exemple, el qual es pot trobar al README és:

java -jar jar/LzDecoder.jar -i "quijote_short.txt" -mode 3 -mDes 8 -mEnt 4

Funcionament

Correccions de l'entrega anterior

Abans de realitzar aquest nou exercici, s'han realitzat una sèrie de modificacions nascudes de la correcció de la pràctica anterior. L'error més important d'aquesta va ser triar el 'espai' per separar les diferents codificacions binàries. Aquest fet suposa l'adició de caràcters en la seqüència i per tant pot donar errors en la compressió i, conseqüentment, en el càlcul del factor de compressió.

El que hem fet és afegir un símbol si tenim el buffer ple. És a dir, tal i com ens van suggerir a classe, per identificar un salt en la codificació, en comptes d'un espai afegim el que fem és inserir un símbol diferent al del buffer quan aquest és ple del mateix símbol. Per exemple, si tenim mDes de mida 6, si ens trobem 5 uns en el buffer, afegim un zero. D'aquesta manera, sempre es pot codificar fent servir el buffer. Aquesta modificació també implica canvis en el descodificador.

Noves implementacions

Tal i com indicava l'enunciat, hem procedit a afegir la classe *textReader* proporcionada per fer ús de les funcions string2ASCIIbin, ASCIIbin2string i *cargarTxt*. La primera, converteix un *String* normal i corrent (en ASCII) a binari. La segona, al revés, li passem un *String* en binari i retorna la conversió en ASCII. La tercera, finalment, llegeix un fitxer de text i retorna el seu contingut en format cadena binaria.

L'ús d'aquesta classe ha suposat les següents modificacions en el nostre codi:

- Canviar d'*String* a *StringBuffer* (per no modificar tot el codi, agafem aquest *StringBuffer* i el convertim a *String*)
- L'abandonament de les nostres pròpies funcions que realitzaven el mateix que les de *txtReader* (segueixen al codi però no es criden mai)

Així doncs, el flux del programa és el següent:

1. Llegim el fitxer amb *cargarTxt* i el guardem a una variable global.
2. Modifiquem aquesta seqüència: si trobem el mateix símbol *mDes* -2 vegades, afegim un nosaltres de diferent. La funció encarregada de realitzar-ho s'anomena *preCode*.
3. Hem creat un mode nou, el 3. Aquest mode serveix per calcular el temps de codificació i descodificació.
4. Comencem a contar amb *System.nanoTime()*
5. Codifiquem l'input
6. Acabem de contar. Guardem el temps de codificació en un variable
7. Comencem a contar de nou

8. Descodifiquem
9. Parem de nou el temps (temps de descodificació)
10. Es comprova que la seqüència inicial concorda amb la traduïda.
11. S'imprimeix per pantalla la mida original, la mida codificada, el factor de compressió i finalment els 100 primers caràcters del text traduït.

Resultats

Un cop el codi adaptat funciona correctament, hem procedit a experimentar amb diferents mides de finestra lliscant i finestra d'entrada. Concretament, tan pel fitxer 'quijote_short' com pel 'hamlet_short' hem realitzat la seva codificació i descodificació, mesurant el temps i calculant per cadascun dels tests el factor de compressió. Aquí estan els resultats:

Fitxer	Mdes	Ment	C. Time	D. Time	Fac. Compres	Total Time
Hamlet	4	4	0,866729877	0,496210257	0,359	1,362940134
Hamlet	8	4	0,290468459	0,161703276	0,524	0,452171735
Hamlet	8	8	0,472772282	0,172658746	0,464	0,645431028
Hamlet	16	4	0,199132548	0,238018453	0,56	0,437151001
Hamlet	16	8	0,175754093	0,150479645	0,56	0,326233738
Hamlet	16	16	0,248639526	0,151223963	0,492	0,399863489
Hamlet	32	4	0,175356335	0,146687343	0,545	0,322043678
Hamlet	32	8	0,136072671	0,162010154	0,645	0,298082825
Hamlet	32	16	0,22318812	0,181939025	0,599	0,405127145
Hamlet	32	32	0,160492081	0,171267072	0,54	0,331759153
Hamlet	64	4	0,25448718	0,151464921	0,499	0,405952101
Hamlet	64	8	0,13409252	0,205053161	0,71	0,339145681
Hamlet	64	16	0,128800063	0,170813955	0,696	0,299614018
Hamlet	64	32	0,220806371	0,147475179	0,639	0,36828155
Hamlet	64	64	0,317670897	0,129225981	0,587	0,446896878
Hamlet	128	4	0,272851258	0,129984057	0,448	0,402835315
Hamlet	128	8	0,225077072	0,348448999	0,708	0,573526071
Hamlet	128	16	0,144994231	0,181051029	0,741	0,32604526
Hamlet	128	32	0,147489899	0,199051908	0,686	0,346541807
Hamlet	128	64	0,239191248	0,167470289	0,635	0,406661537
Hamlet	128	128	0,626331594	0,103375217	0,593	0,729706811
Hamlet	256	4	0,454759243	0,116620917	0,408	0,57138016
Hamlet	256	8	0,146892782	0,135361313	0,694	0,282254095
Hamlet	256	16	0,180932309	0,197621515	0,792	0,378553824
Hamlet	256	32	0,169661	0,182809101	0,747	0,352470101
Hamlet	256	64	0,440430667	0,241628038	0,698	0,682058705
Hamlet	256	128	0,624142164	0,144691832	0,657	0,768833996
Hamlet	256	256	1,887489784	0,068708492	0,621	1,956198276
Hamlet	512	4	0,49550114	0,168952203	0,379	0,664453343
Hamlet	512	8	0,282095056	0,063927394	0,671	0,34602245
Hamlet	512	16	0,25217231	0,0847049	0,859	0,33687721
Hamlet	512	32	0,168145486	0,137227545	0,849	0,305373031

Hamlet	512	64	0,278147874	0,110825423	0,809	0,388973297
Hamlet	512	128	0,633566441	0,165782937	0,769	0,799349378
Hamlet	512	256	1,634698276	0,065999064	0,73	1,70069734
Hamlet	512	512	8,126684553	0,159589365	0,698	8,286273918
Hamlet	1024	4	1,107778397	0,090629994	0,363	1,198408391
Hamlet	1024	8	0,552020887	0,121155937	0,643	0,673176824
Hamlet	1024	16	0,373066809	0,135394273	0,93	0,508461082
Hamlet	1024	32	0,320318885	0,129009022	0,999	0,449327907
Hamlet	1024	64	0,393487837	0,10724336	0,986	0,500731197
Hamlet	1024	128	0,579388764	0,106202084	0,96	0,685590848
Hamlet	1024	256	1,480881686	0,075443182	0,911	1,556324868
Hamlet	1024	512	5,308672217	0,124103444	0,883	5,432775661
Hamlet	1024	1024	28,003574384	0,40624394	0,855	28,40981832
Hamlet	2048	4	2,124175764	0,180170393	0,371	2,304346157
Hamlet	2048	8	1,140506571	0,105023849	0,639	1,24553042
Hamlet	2048	16	0,876074795	0,06040101	0,961	0,936475805
Hamlet	2048	32	0,690171308	0,155506824	1,094	0,845678132
Hamlet	2048	64	0,670264518	0,050966491	1,102	0,721231009
Hamlet	2048	128	0,776954919	0,105922405	1,09	0,882877324
Hamlet	2048	256	1,565562587	0,083354187	1,044	1,648916774
Hamlet	2048	512	3,997836169	0,123811605	1,017	4,121647774
Hamlet	2048	1024	19,284232647	0,548306344	0,972	19,83253899
Hamlet	2048	2048	-	-	-	-
Hamlet	4096	4	4,162148714	0,08031676	0,441	4,242465474
Hamlet	4096	8	2,38169237	0,054836875	0,693	2,436529245
Hamlet	4096	16	1,569858887	0,065354907	0,967	1,635213794
Hamlet	4096	32	1,165965976	0,053820559	1,133	1,219786535
Hamlet	4096	64	1,064383392	0,065577946	1,192	1,129961338
Hamlet	4096	128	1,105757286	0,065300188	1,202	1,171057474
Hamlet	4096	256	1,281950737	0,064920669	1,17	1,346871406
Hamlet	4096	512	2,641246407	0,136438748	1,151	2,777685155
Hamlet	4096	1024	10,317774576	0,509555797	1,094	10,82733037
Hamlet	4096	2048	45,733224235	4,975233711	1,026	50,70845795

Mentre que pel fitxer del Quijote...:

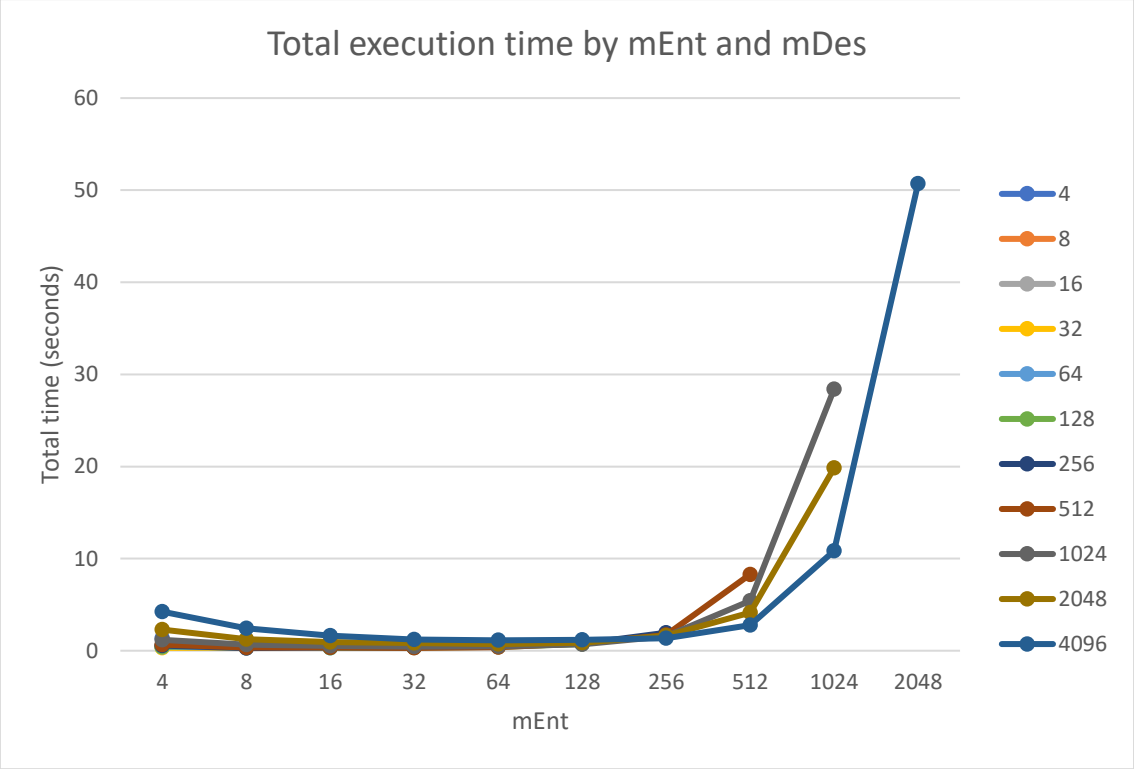
Fitxer	Mdes	Ment	C. Time	D. Time	Fac. Compres	Total Time
Quijote	4	4	0,908201371	0,53343249	0,341	1,441633861
Quijote	8	4	0,297787786	0,112267337	0,506	0,410055123
Quijote	8	8	0,351598105	0,124001684	0,443	0,475599789
Quijote	16	4	0,201112059	0,259463797	0,562	0,460575856
Quijote	16	8	0,242637313	0,185768448	0,569	0,428405761
Quijote	16	16	0,185930367	0,188114037	0,504	0,374044404
Quijote	32	4	0,22320764	0,197190797	0,551	0,420398437
Quijote	32	8	0,137863382	0,158106812	0,654	0,295970194
Quijote	32	16	0,161560556	0,165579738	0,617	0,327140294

Quijote	32	32	0,186104126	0,249233123	0,557	0,435337249
Quijote	64	4	0,186208446	0,204828843	0,499	0,391037289
Quijote	64	8	0,151637721	0,263035941	0,71	0,414673662
Quijote	64	16	0,152717715	0,258494202	0,718	0,411211917
Quijote	64	32	0,154623307	0,154363148	0,663	0,308986455
Quijote	64	64	0,222086365	0,223072281	0,61	0,445158646
Quijote	128	4	0,24778609	0,229309693	0,448	0,477095783
Quijote	128	8	0,138790098	0,165984537	0,714	0,304774635
Quijote	128	16	0,117367794	0,173967541	0,795	0,291335335
Quijote	128	32	0,137634264	0,203553648	0,747	0,341187912
Quijote	128	64	0,362906855	0,161215597	0,692	0,524122452
Quijote	128	128	0,667767568	0,107542878	0,645	0,775310446
Quijote	256	4	0,415331259	0,195083286	0,408	0,610414545
Quijote	256	8	0,195266645	0,178213281	0,7	0,373479926
Quijote	256	16	0,137181465	0,147513259	0,866	0,284694724
Quijote	256	32	0,146441583	0,208743385	0,842	0,355184968
Quijote	256	64	0,200444542	0,173481142	0,792	0,373925684
Quijote	256	128	0,604243373	0,197301836	0,742	0,801545209
Quijote	256	256	1,628113346	0,100582269	0,7	1,728695615
Quijote	512	4	0,435037091	0,14507135	0,379	0,580108441
Quijote	512	8	0,297218188	0,066007064	0,671	0,363225252
Quijote	512	16	0,17867568	0,112844934	0,927	0,291520614
Quijote	512	32	0,155848582	0,120019302	0,962	0,275867884
Quijote	512	64	0,215653434	0,130406456	0,917	0,34605989
Quijote	512	128	0,530489464	0,147697258	0,867	0,678186722
Quijote	512	256	0,530489464	0,147697258	0,867	0,678186722
Quijote	512	512	5,687137722	0,120323301	0,784	5,807461023
Quijote	1024	4	0,90190796	0,079545884	0,364	0,981453844
Quijote	1024	8	0,586335293	0,073378551	0,644	0,659713844
Quijote	1024	16	0,362753895	0,148123497	0,967	0,510877392
Quijote	1024	32	0,323178712	0,053020562	1,078	0,376199274
Quijote	1024	64	0,384618117	0,068385294	1,046	0,453003411
Quijote	1024	128	0,548755621	0,157136256	0,993	0,705891877
Quijote	1024	256	1,776309162	0,1070296	0,942	1,883338762
Quijote	1024	512	5,218308142	0,131478131	0,907	5,349786273
Quijote	1024	1024	27,35292674	0,539804941	0,877	27,89273168
Quijote	2048	4	2,043612605	0,130781495	0,372	2,1743941
Quijote	2048	8	1,205594919	0,058568377	0,639	1,264163296
Quijote	2048	16	0,911382157	0,054737674	0,981	0,966119831
Quijote	2048	32	0,672247549	0,122337052	1,13	0,794584601
Quijote	2048	64	0,059841012	0,744956343	1,112	0,804797355
Quijote	2048	128	1,295526996	0,117328434	1,067	1,41285543
Quijote	2048	256	2,061558124	0,094218458	1,022	2,155776582
Quijote	2048	512	4,881399891	0,103181298	0,985	4,984581189
Quijote	2048	1024	20,52354165	0,60276402	0,954	21,12630567
Quijote	2048	2048	-	-	-	-

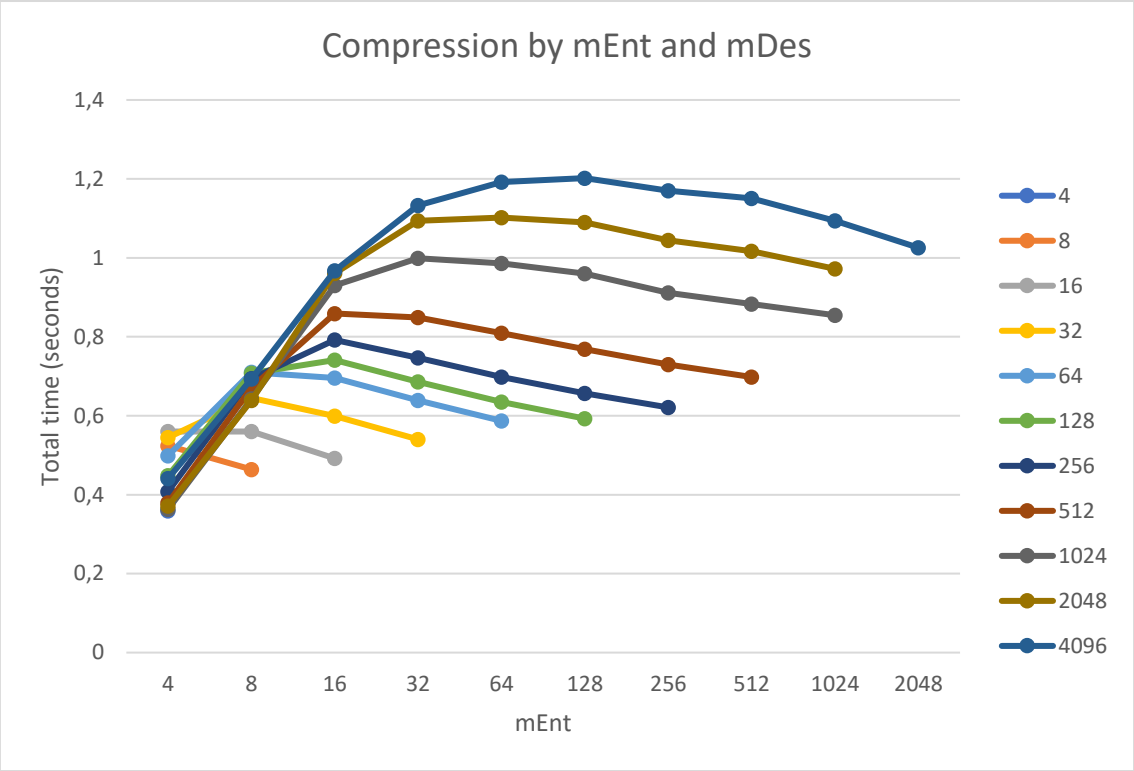
Quijote	4096	4	4,085068099	0,129376701	0,445	4,2144448
Quijote	4096	8	2,564095392	0,072845434	0,696	2,636940826
Quijote	4096	16	1,795742355	0,06692002	0,978	1,862662375
Quijote	4096	32	1,310074451	0,09586901	1,11	1,405943461
Quijote	4096	64	1,227793539	0,057443583	1,105	1,285237122
Quijote	4096	128	1,282903693	0,063318757	1,079	1,34622245
Quijote	4096	256	1,699487106	0,07588702	1,05	1,775374126
Quijote	4096	512	3,268388558	0,131258292	1,029	3,39964685
Quijote	4096	1024	12,50864732	0,58040892	1,006	13,08905624
Quijote	4096	2048	53,83605834	5,587662008	0,994	59,42372034

Gràcies a aquestes taules hem pogut generar els gràfics demanats. A continuació els exposem amb les nostres conclusions.

Hamlet

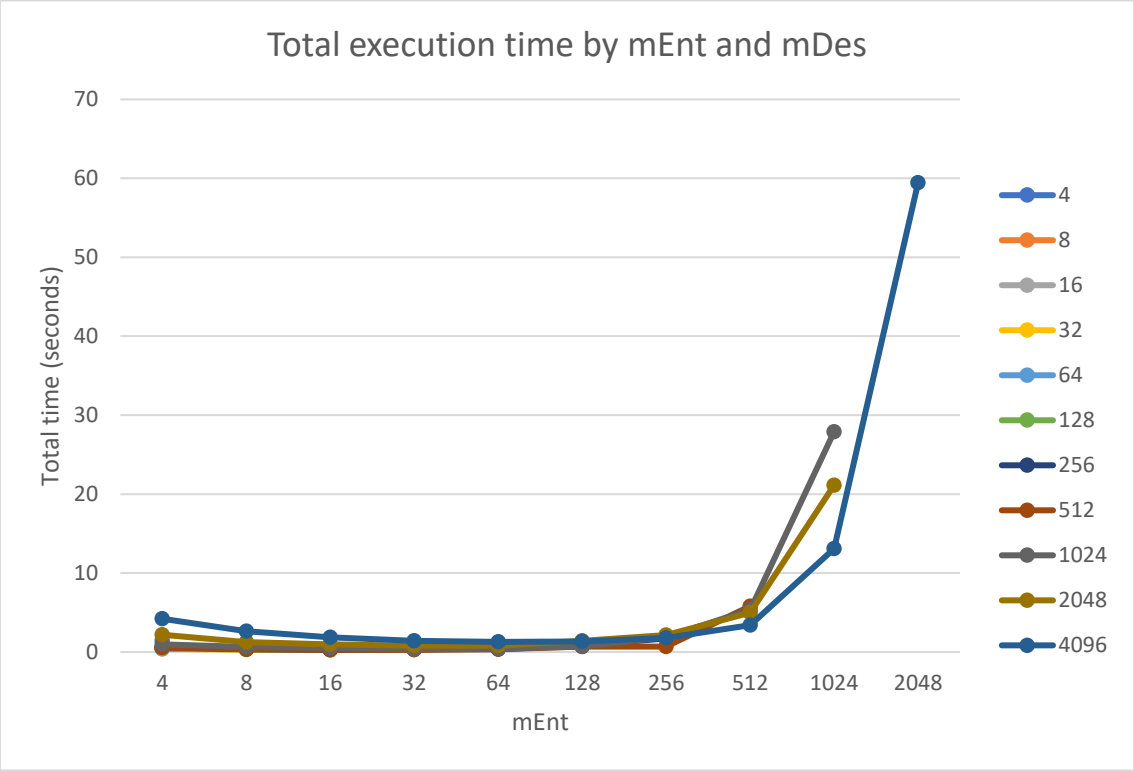


En aquest gràfic es mostra el temps total de codificació i descodificació en comparació a la mida del buffer d'entrada (*mEnt*, eix de les x) i de la mida del buffer lliscant (*mDes*, eix de les y). El menor dels temps es troba a *mDes* = 256 i *mEnt* = 8, amb un temps de 0,2822 segons i un factor de compressió de 0,69. El temps més gran el trobem a on *mDes* i *mEnt* són més grans: 4096 i 2048 respectivament. Això té tot el sentit del món ja que cal omplir buffers molt grans, i per tant la cerca d'elements dins d'aquests per a la codificació és molt més gran.

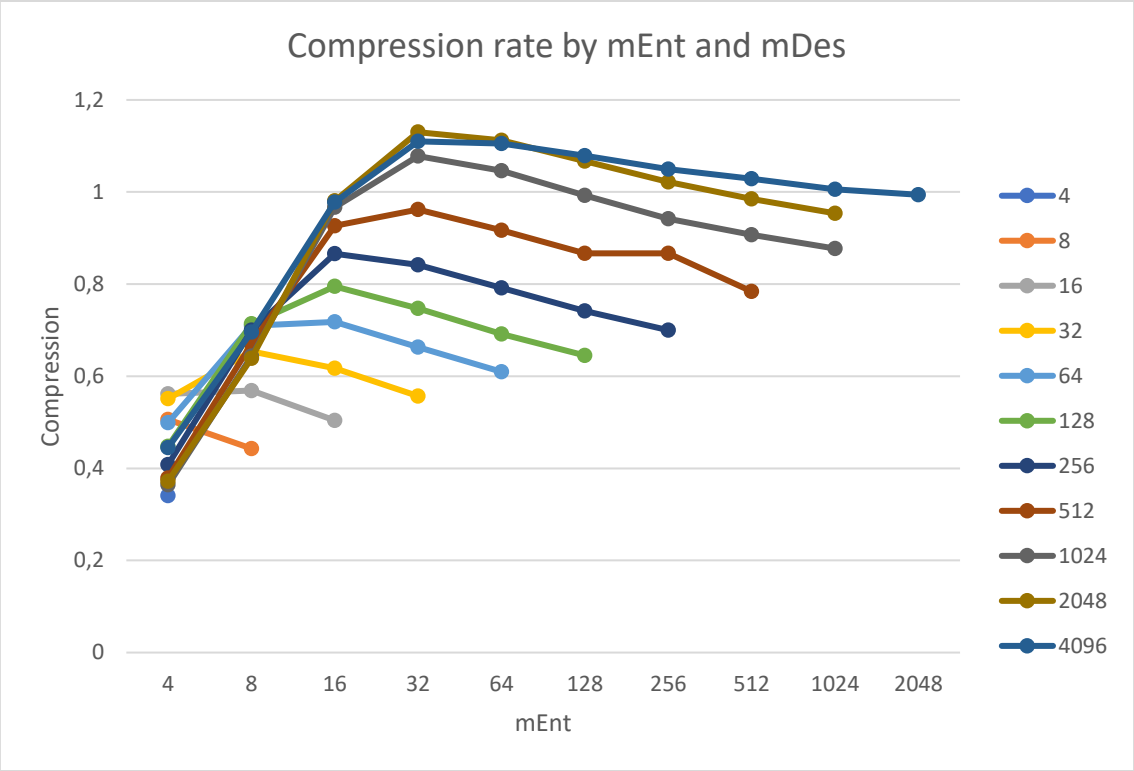


Aquest altre gràfic mostra el factor de compressió en relació al buffer lliscant (*mDes*, eix de les y) i el d'entrada (*mEnt*, eix de les x). El major dels factors és de 1,202, el qual s'aconsegueix amb *mDes* = 4096 i *mEnt* = 128. En canvi, el menor dels factors és de 0,359 i òbviament s'aconsegueix amb els buffers *mEnt* i *mDes* de mida mínima, és a dir, 4.

<u>Quijote</u>



Aquest gràfic és igual al primer però pel fitxer 'quijote_short'. En aquest cas, el temps mínim és de 0,275 i s'aconsegueix amb *mEnt* = 32 i *mDes* = 512. El màxim, de 59 segons, s'aconsegueix on *mEnt* i *mDes* són més grans (igual que en el fitxer anterior), és a dir, 2048 i 4096 respectivament.



Finalment, aquest gràfic ens mostra que el major dels factors de compressió pel segon fitxer és de 1,13 i s'aconsegueix amb *mDes* = 2048 i *mEnt* = 32. El mínim, un altre cop, s'aconsegueix on *mDes* i *mEnt* són mínims.

<u>Conclusions</u>

En el fitxer Hamlet aconseguim major compressió ja que és un text d'una obra de teatre, on s'escriu en cada frase el nom del personatge que la pronuncia. En conseqüència, aquest *String* es repeteix nombroses vegades en tot el text, aconseguint així una codificació més eficaç. En el fitxer del Quijote també és repeteixen paraules (sobretot preposicions, pronoms o determinants, ja que el text és en castellà i no en anglès), però no tantes ja que no s'escriu el nom dels personatges cada vegada que parlen.

En termes de temps, codificar i descodificar els dos fitxers requereix més o menys el mateix temps. S'ha de tenir en compte que el temps d'execució és d'una sola execució, al ser executat en un PC d'usuari, pot ser que els temps hagin variat depenent de la gestió de recursos del sistema operatiu (per a millorar la veracitat dels tests s'haurien de fer múltiples proves per a cada valor). Tot i això podem dir que el text de Hamlet es comprimeix més ràpidament que el de quijote, ja que al haver-hi paraules repetides no es fa una búsqueda completa de patrons dins el buffer. Ho podem veure en els tests amb mDes=4096 i mEnt=2048, on hi ha una diferència de quasi 10 segons.

El fitxer del quijote té més paraules i està en castellà, mentres que el de Hamlet està en anglès.

Hi ha tests que no hem pogut realitzar degut al seu alt cost computacional. Són aquells marcats amb un guió a les taules de valors.