

بسمه تعالی



تمرین کامپیوتری شماره ۱ درس مخابرات دیجیتال

استاد درس: آقای دکتر بهنیا

تمرین اول کامپیوتری، شامل ۲ سوال مربوط به بخش کدگذاری منبع می باشد.

لطفاً پیش از اتمام موعد تحویل، گزارش خود را در سامانه CW بارگذاری نمایید. عنوان فایل بارگذاری شده در سامانه را به

صورت DigitCom\_CA1\_student number انتخاب نمایید. فایل ارسالی شما، بصورت یک فایل zip و شامل گزارش به

فرمت PDF همراه با کد های متلب مورد استفاده برای هر مورد می باشد. تمرینی که در سامانه CW و در موعد مقرر بارگذاری

نشود، تصحیح نخواهد شد. در صورت داشتن سوال، می توانید از طریق ایمیل زیر با من در ارتباط باشید.

ایمیل: [mohammadreza.shamsian@ee.sharif.edu](mailto:mohammadreza.shamsian@ee.sharif.edu)

❖ لطفا تمرین خود را به ایمیل ارسال ننمایید زیرا که تصحیح نخواهد شد.

❖ قسمت عمده نمره شما را تحلیل و توصیف نتایجی که بدست آورده اید تشکیل خواهد داد.

❖ هرگونه مشابهت در کد یا گزارش، منجر به صفر شدن نمره این تمرین خواهد شد.

موفق باشید

شمسیان

در سوالات زیر، اگر نیاز به فرض هایی داشتید که در صورت سوال بیان نشده بود، می توانید آن موارد را خودتان در نظر بگیرید اما در گزارش، مفروضات خود را بیان نمایید.

با استفاده از نرم افزار متلب، تابع `MyHuffman.m` را برای کدگذاری هافمن بنویسید. این تابع باید کاراکتر های ورودی را دریافت کرده و با شمارش تعداد آنها در داده ورودی، احتمال وقوع هر سمبل را محاسبه نماید و در نهایت، کد باینری هافمن متناظر با آنها را برگرداند. برای سوال اول و برای سادگی می توانید به هر سمبل ورودی، یک عدد دو رقمی (مثلا از ۱۰ تا ۴۲) نسبت دهید. برای سوال دوم نیز، از دستوراتی که معرفی شده اند استفاده کرده و هر سمبل را با اعدادی که پس از خواندن فایل مشاهده میکنید، معادل بگیرید. همچنین تابع `MyLempelZiv.m` را برای کدگذاری `Lempel Ziv` بنویسید. این تابع نیز باید سمبل های ورودی را دریافت کرده و کد باینری `Lempel Ziv` آنها را برگرداند.

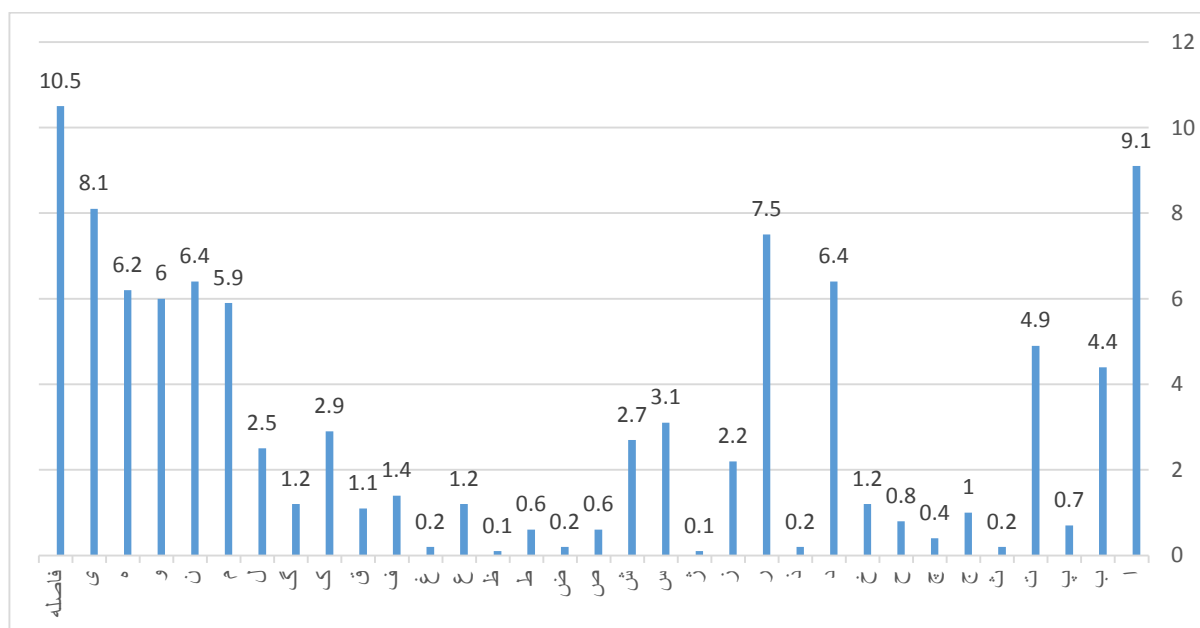
در سوالات زیر، نرخ فشرده سازی را از رابطه زیر محاسبه کنید:

$$CR = \frac{\text{original data size}}{\text{compressed data size}}$$

سوال اول:

با توجه به جدول فراوانی الفبای فارسی زیر (اعداد روی ستون ها در صد وقوع حروف می باشند) متنی فارسی بصورت تصادفی و به طول ۲۰۰۰ کاراکتر تولید کرده، با استفاده از توابعی که برای سوال ۱ نوشته اید، آن را کد کنید (هر دو نوع کد).

برای کد هافمن، و برای این سوال، می توانید خروجی تابع یا بخشی که به محاسبه احتمال کاراکتر ها می پردازد را با نمودار زیر که در آن احتمالات وقوع مشخص است، جایگزین نمایید.



۱) نرخ فشرده سازی برای هر دو نوع کد چقدر است؟ (در صورتی که کدگذاری های هافمن و `Lempel-Ziv` استفاده نشوند، هر سمبل با ۶ بیت کد می شود)

۲) در گیرنده، برای کدگشایی، به دیکشنری نیاز خواهید داشت. یعنی باید مشخص باشد هر سمبل با چه کلمه کدی، کدگذاری شده است. در نتیجه میزان حافظه مورد نیاز درگیرنده مهم خواهد بود. دیکشنری هایی که برای کدگشایی لازم دارید، برای کدهای Huffman و Lempel Ziv و کد ثابت به طول ۶ بیت، هر کدام به چند بیت نیاز دارند؟

۳) در کدگذاری Huffman، احتمال مربوط به حروف چگونه تغییر کند تا میزان فشردگی حداکثر شود؟ میزان فشردگی سازی در این حالت چقدر است؟

۴) با استفاده از دستور زیر، فایل متنی دوم را بخوانید و رشته بیت تولید شده در متغیر dataInBinary را بعنوان رشته ورودی برای هر دو تابع کدگذاری در نظر بگیرید. مقدار نرخ فشردگی سازی برای هر کد چقدر است؟ کدام کد عملکرد بهتری دارد و چرا؟

```
- fid = fopen('...direction of your files\file2.txt');
- dataInDecimal = fread(fid);
- temp = de2bi(dataInDecimal);
- dataInBinary = reshape(temp.', 1, numel(temp));
- fclose(fid);
```

#### سوال دوم:

هدف از این سوال، مقایسه کدینگ های Huffman و Lempel Ziv از نظر فشردگی سازی و زمان کدگذاری می باشد. با توجه به توابعی که در بالا نوشته اید، برای هر دو نوع کد معرفی شده، دو دسته فایلی که در اختیار شما قرار داده شده است را کد کنید و سپس جداول زیر را تکمیل نمایید. دو دسته فایلی که در ضمیمه در اختیار شما هستند، فایل های متن و تصویر می باشند. برای محاسبه زمان اجرای کد، می توانید از دستور tic-toc در متلب استفاده نمایید. با توجه به نحوه پیاده سازی هر الگوریتم و سیستم مورد استفاده، می تواند زمان بدست آمده برای افراد مختلف، متفاوت باشد. پس سعی کنید الگوریتم ها را بصورت بهینه پیاده سازی کنید تا بتوانید مقایسه درستی در مورد دو الگوریتم داشته باشید.

الف) کدگذاری متنی:

فایل های موجود در پوشه Text با فرمت txt و بصورت ASCII ذخیره شده اند. برای خواندن آنها، می توانید از دستور زیر استفاده کنید.

```
- fid = fopen('...direction of your files\file1.txt');
- txt = fread(fid);
- fclose(fid);
```

برای محاسبه نرخ فشردگی سازی، برای حالتی که از کدگذاری های هافمن یا Lempel Ziv استفاده نمی کنید، تعداد بیت لازم برای کد کردن سمبل های ورودی را بدست آورید.

File name	Number of characters	Number of symbols	
		Huffman	Lempel-Ziv
Text file 1	1456		
Text file 2	4674		
Text file 3	9010		

Text file	Original number of bits	Huffman			Lempel-Ziv		
		Compressed number of bits	Compression ratio (%)	Compression time (sec)	Compressed number of bits	Compression ratio (%)	Compression time (sec)
1							
2							
3							
		Mean					
		Standard deviation					

پس از تکمیل جداول بالا، به سوالات زیر پاسخ دهید:

- (۱) برای کدگذاری فایل های متنی، کدام یک از دو نوع کد بالا را پیشنهاد می دهید و چرا؟  
 (۲) کد Lempel-Ziv برای متن های با طول بزرگ مناسب تر است یا متن های با طول کمتر؟ چرا؟

(ب) کدگذاری تصویر:

آن دسته از فایل های موجود در پوشه Image با فرمت jpg، بصورت uint8 ذخیره شده اند. برای خواندن و نمایش تصاویر، می توانید از دستورات زیر استفاده کنید. ماتریس اعداد مربوط به تصاویر را خوانده و سپس آن را در یک بردار بریزید (ستون های ماتریس را زیر هم بچینید).

```
- pic = imread('...direction of your files\image1.jpg');
- imshow(pic);
```

File name	Number of layers	Number of symbols	
		Huffman	Lempel-Ziv
Image file 1	1		
Image file 2	1		
Image file 3	1		
Image file 4	1		
Image file 5	1		

image file	Original number of bits	Huffman			Lempel-Ziv		
		Compressed number of bits	Compression ratio (%)	Compression time (sec)	Compressed number of bits	Compression ratio (%)	Compression time (sec)
1							
2							
3							
4							
5							
		Mean					
		Standard deviation					

پس از تکمیل جداول بالا، به سوالات زیر پاسخ دهید:

(۱) برای تصاویر مختلف، نتایج مربوط به کد Huffman را تحلیل کنید. همین کار را برای کد Lempel-Ziv انجام دهید.

(۲) دو نوع کد Huffman و Lempel Ziv را از نظر نرخ فشرده سازی و زمان فشرده سازی با هم مقایسه کنید و دلایل تفاوت آنها را بنویسید. با توجه به نتایج بالا، شما کدام نوع کد را برای فشرده سازی تصویر پیشنهاد می دهید؟ علت انتخاب خود را توضیح دهید.