به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

# تمرین کامپیوتری ۱

## مخابرات دیجیتال

### دکتر ربیعی

**عرفان پناهی ۸۱۰۱۹۸۳۶۹**

نیمسال دوم ۰۱-۱۴۰۰

# فهرســـت:

*** فایل مربوط به این تمرین کامپیوتری با نام **CA1_panahi.m** پیوست شده است.

## سوال ۱۰ : محاسبه $G_k$ با استفاده از ماتریس احتمالات گذار

تعریف $G_k$ به صورت زیر است:

$$G_k \triangleq \frac{H(X_1, X_2, \dots, X_k)}{k}$$

از طرفی می‌دانیم:

$$\begin{aligned} H(X_1, X_2, \dots, X_k) &= H(S_1, S_2, \dots, S_k) - H(S_0 | X_1, X_2, \dots, X_k) \\ &= H(S_0) + H(S_1|S_0) + H(S_2|S_1, S_0) + \dots + H(S_k|S_{k-1}, \dots, S_0) \\ &- H(S_0|X_1, X_2, \dots, X_k) \end{aligned}$$

با توجه به مارکوف بودن منبع داریم:

$$H(X_1, X_2, \dots, X_k) = H(S_0) + H(S_1|S_0) + H(S_2|S_1) + \dots + H(S_k|S_{k-1}) - H(S_0|X_1, X_2, \dots, X_k)$$

همچنین از ایستان بودن منبع نیز داریم:

$$H(X_1, X_2, \dots, X_k) = H(S_0) + kH(S_1|S_0) - H(S_0|X_1, X_2, \dots, X_k)$$

در نتیجه $G_k$ را می‌توان بصورت زیر بدست آورد:

$$G_k = \frac{H(S_0) + kH(S_1|S_0) - H(S_0|X_1, X_2, \dots, X_k)}{k}$$

با توجه به اینکه $H(S_0|X_1, X_2, \dots, X_k) \leq H(S_0)$ ، میتوان از این ترم صرف نظر کرد. در صورتی که $k$ به سمت بینهایت میل کند داریم:

$$\lim_{k\to\infty} G_k = \lim_{k\to\infty} \frac{H(S_0) + kH(S_1|S_0) - H(S_0|X_1, X_2, \dots, X_k)}{k} = H(S_1|S_0)$$

پس رابطه زیر را برای محاسبه $G_k$ دنبال می‌کنیم. باید توجه کرد که به ازای $k$ های کوچک ، حاصل بدست آمده دقت ندارد اما برای $k$ های بزرگ پاسخ بدست آمده صحیح است.

$$G_k \cong \frac{H(S_0) + kH(S_1|S_0)}{k}$$

تابع *entropy.m* ، برای محاسبه $G_k$ نوشته شده است. در این تابع ابتدا بردار حالت اولیه (P) را بدست می‌آوریم. با توجه به ایستان بودن منبع می‌توان گفت:

$$\Phi^T P = P$$

از طرفی : $sum(P) = 1$

می‌توانیم معادله بالا را با استفاده از دستور *solve* متلب حل کنیم. راه دیگر استفاده از مقدار و بردار ویژه است. در حقیقت باید بردار ویژه متناظر با مقدار ویژه ۱ را پیدا کنیم تا معادله $\Phi^T P = P$ ارضا شود. سپس از معادله دوم استفاده کرده و $P$ بدست آمده را نرمالیزه می‌کنیم ($P = \frac{P}{sum(P)}$).

پس با استفاده از دستور eig() ، بردار ها و مقادیر ویژه ماتریس $\Phi^T$ را بدست می‌آوریم. سپس بردار ویژه متناظر با مقدار ویژه ۱ را پیدا می‌کنیم و آنرا نرمالیزه می‌کنیم.

برای محاسبه $H(S_0)$ از بردار حالت اولیه $P$ استفاده می‌کنیم:

$$H(S_0) = -\sum_i P_i \log_2(P_i)$$

برای محاسبه $H(S_1|S_0)$ ابتدا باید $H_i$ ها را بدست آوریم. به این منظور باید آنتروپی را برای هر سطر $\Phi_{n\times n}$ بدست می‌آوریم:

$$H_i = -\sum_{j=1}^{n} p_{ij} \log_2(p_{ij})$$

$$H(S_1|S_0) = \sum_{i=1}^{n} P_i H_i$$

در نهایت نیز با جایگذاری دو آنتروپی به دست آمده در رابطه زیر، $G_k$ بدست می‌آید.

$$G_k \cong \frac{H(S_0) + kH(S_1|S_0)}{k}$$

بعنوان نمونه یکی از مثال های جزوه درس را بررسی می‌کنیم. تصویر۱ مقدار محاسبه شده $G_k$ برای $k = 1,2,3$ را به همراه مقدار دقیق آن نشان میدهد. همانطور که مشاهده با افزایش $k$ ، تقریب زده شده بهتر می‌شود و دقت بیشتری در محاسبه $G_k$ خواهیم داشت.

```
Problem 10:
  test case: ts = [3/4 1/4;1/4 3/4]
  G1 = 1.811278 (ex:1.56) , G2 = 1.311278 (ex:1.28), G3 = 1.144611 (ex:1.136)
```

تصویر ۱ : بررسی صحت عملکرد تابع entropy.m

## سوال ۱۱ : محاسبه متوسط طول کد هافمن

برای محاسبه متوسط طول کد هافمن، مطابق با توضیحات صورت پروژه، ابتدا رشته سمبل دریافت شده را به کلمات $k$ تایی تبدیل می‌کنیم. به این منظور کلمه اول را بصورت حروف $1$ تا $k$ ، کلمه دوم حروف $2$ تا $k+1$ و ... تعریف می‌کنیم. در نتیجه در پایان $n-k+1$ تا کلمه $k$ تایی داریم. در مرحله بعد باید کلمات یکتا را تشخیص داده و احتمال هر کدام را بدست آوریم.

برای اینکه بتوانیم، سمبل های یکتا را در بین کلمات بدست آمده پیدا کنیم از دستور ***unique*** استفاده می‌کنیم. اولین خروجی این تابع، کلمات یکتا هستند. سومین خروجی این تابع نیز کلمات برحسب اندیس های کلمات یکتا هستند.

با استفاده از خروجی سوم فراوانی هر کلمه را بدست آورده و سپس تقسیم بر کل کلمات ($n-k+1$) می‌کنیم تا احتمال هر کلمه یکتا بدست آید. سپس این احتمالات را به همراه دنباله سمبل های یکتا به دستور ***huffmandict()*** می‌دهیم و خروجی که متوسط طول کد هافمن است را بدست می‌آوریم.

بعنوان نمونه رشته *chain = ‘AABABB’* را به تابع می‌دهیم. برای $k=1$ ، چون صرفا دو سمبل *A* و *B* داریم، متوسط طول کد هافمن ۱ خواهد بود. برای $k=2$ و $k=3$ ، متوسط طول کد هافمن ۲ خواهد بود. تصویر ۲ نتایج این مثال را نشان می‌دهد.

```
Problem 11:
  test case: chain = 'AABABB'
  n1 = 1.000000 , n2 = 2.000000 , n3 = 2.000000
```

تصویر ۲ : بررسی صحت عملکرد تابع average_length.m

## سوال ۱۲ : محاسبه و رسم متوسط طول کد هافمن ، $G_k$ و بهره کدینگ برای یک منبع با حافظه

**محاسبه نرخ آنتروپی به صورت تحلیلی :** برای منبع با حافظه داده شده ، ماتریس احتمالات گذار به صورت زیر تعریف می‌شود:

$$\Phi = \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 0.8 & 0.2 \end{bmatrix}$$

حال بردار حالت اولیه را بدست می‌آوریم:

$$\begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.8 \\ 0.5 & 0.2 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \rightarrow P_1 = 0.5P_1 + 0.8P_2 \rightarrow P_1 = 1.6P_2$$

$$P_1 + P_2 = 1 \rightarrow 2.6P_2 = 1 \rightarrow P_2 = \frac{5}{13}\ , P_1 = \frac{8}{13}$$

در نتیجه $H_i$ ها به صورت زیر محاسبه می‌شوند:

$$H_1 = -(1-\alpha)\log_2(1-\alpha) - \alpha\log_2\alpha = h_b(\alpha) = h_b(0.5) = 1$$

$$H_2 = -\beta\log_2\beta - (1-\beta)\log_2(1-\beta) = h_b(\beta) = h_b(0.8) = 0.7219$$

در نهایت نیز نرخ آنتروپی منبع بی حافظه به صورت زیر محاسبه می‌شود:

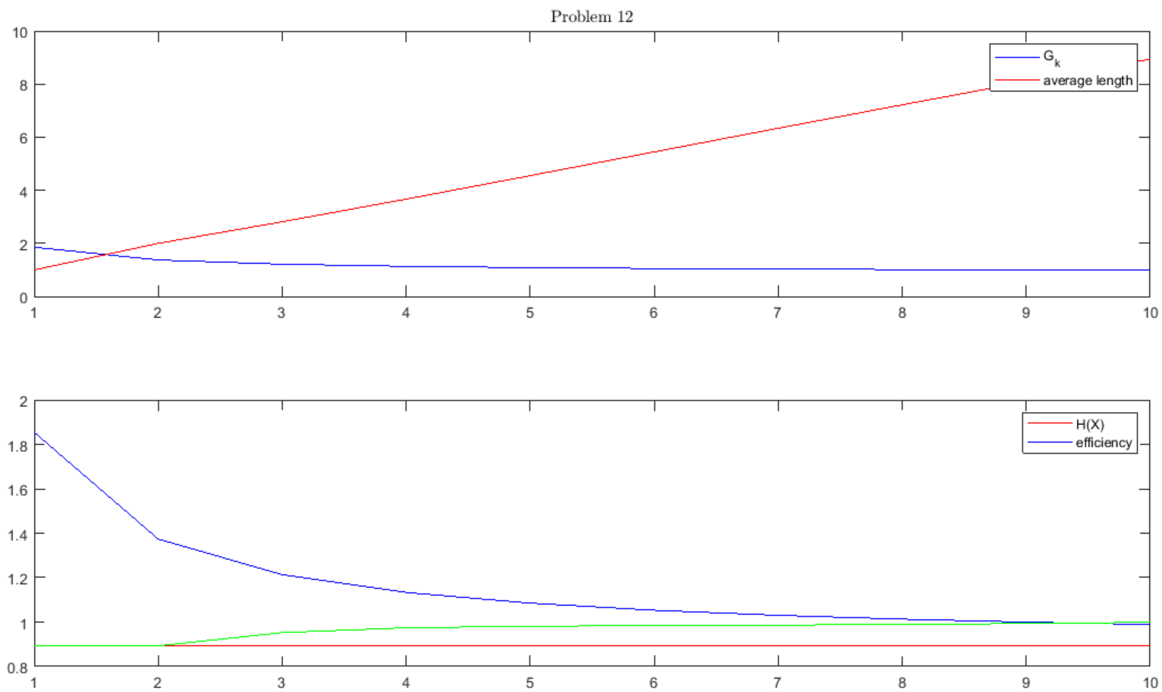
$$H(\mathbb{X}) = H_1P_1 + H_2P_2 = \frac{8}{13} \times 1 + \frac{5}{13} \times 0.7219 = 0.893$$

**تولید سمبل منبع با حافظه :** برای اینکه سمبل هایی از منبع با حافظه تولید کنیم، ابتدا با استفاده از دستور ***cumsum*** و بردار حالت اولیه، اولین استیت را تعیین می‌کنیم. به این منظور یک عدد تصادفی یکنواخت، بین 0 و 1 تولید می‌کنیم و در صورتی که عدد تصادفی تولید شده در بازه 0 تا $P_1$ باشد، وارد استیت اول شده و در صورتی که در بازه $P_1$ تا 1 باشد حالت اولیه استیت دوم خواهد بود. حال در هر استیت دو احتمال (رفتن به استیت بعدی) وجود دارد که هر کدام سمبل A یا B را تولید می‌کنند. پس از هر انتقال، استیت ذخیره شده و مرحله بعد احتمالات آن استیت را در نظر می‌گیریم (منظور از احتمالات هر استیت ، سطر مربوط به هر استیت در ماتریس احتمالات گذار است.). با تولید عدد تصادفی دیگر استیت بعدی و سمبل تولید شده مشخص می‌شود.

**محاسبه متوسط طول کد هافمن ، $G_k$ و بهره کدینگ با استفاده از *MATLAB* :** با استفاده از توابع دو بخش قبلی به ازای هر $k$ ، متوسط طول کد هافمن ، $G_k$ را بدست می‌آوریم. سپس بهره کدینگ را با استفاده از رابطه زیر محاسبه می‌کنیم:

$$\hat{H}_k = \frac{\bar{n}}{k} \quad , \quad \eta_k = \frac{H(X)}{\hat{H}_k}$$

تصویر ۳ نمودار های خواسته شده در این مسئله را نشان می‌دهد. همانطور که در تصویر نیز مشاهده می‌شود، با افزایش $k$ ، $G_k$ به $H(X)$ میل می‌کند. همچنین با افزایش طول سمبل های ارسالی از کانال مشاهده می‌شود که بهره کدینگ روند صعودی داشته و به ۱ میل می‌کند. تحلیل دقیق تر این مسئله در سوال ۱۴ به طور مفصل تری انجام می‌شود.




تصویر ۳ : متوسط طول کد هافمن ، $G_k$ و بهره کدینگ برای یک منبع با حافظه

## سوال ۱۳ : محاسبه پارامتر های منابع بی حافظه $X^k$

**تولید رشته برای منبع $X^1$** : ابتدا متناظر با احتمالات تولید سمبل های داده شده، یک بردار احتمال تعریف کرده و سمبل ها را از ۱ تا ۳ نامگذاری می‌کنیم. سپس ماتریس احتمالات گذار را برای منبع تعریف می‌کنیم. به این منظور که انتقال به هر استیت باید یک احتمال مجزا داشته باشد (سطر های ماتریس احتمالات یکسان است.). سپس با استفاده از دستور *randsrc* یک رشته به طول ۱۰۰۰ تولید می‌کنیم و آنرا به رشته کاراکتری تبدیل می‌کنیم.

**تولید رشته برای منبع $X^2$** : با استفاده از بردار احتمال قسمت قبل و دستور *kron* ، بردار احتمال برای سمبل های منبع $X^2$ که ۹ تا هستند تولید می‌کنیم. سپس مشابه قسمت قبل یک رشته به طول ۱۰۰۰ تولید می‌کنیم و به کاراکتر تبدیل می‌کنیم.

**تولید رشته برای منبع $X^3$** : برای این منبع، با استفاده از بردار های احتمال دو قسمت قبل دستور *kron* ، بردار احتمال برای سمبل های منبع $X^3$ که ۲۷ تا هستند تولید می‌کنیم. سپس مشابه قسمت قبل یک رشته به طول ۱۰۰۰ تولید می‌کنیم و به کاراکتر تبدیل می‌کنیم.

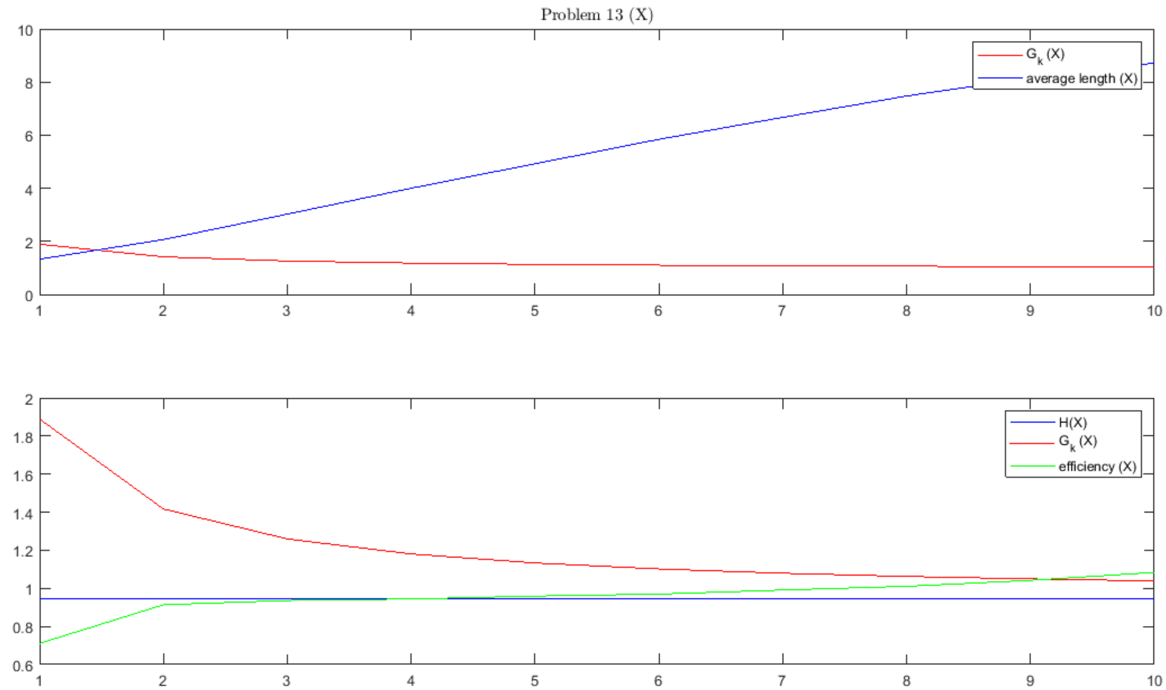
سپس با استفاده از رشته های تولید شده، متوسط کد هافمن را بدست می‌آوریم. همچنین $G_k$ را نیز با استفاده از ماتریس گذار هر منبع محاسبه می‌کنیم.

با استفاده از بردار احتمالات نیز ، آنتروپی هر منبع را بدست می‌آوریم. تصویر ۴ آنتروپی هر سه منبع را نشان می‌دهد.

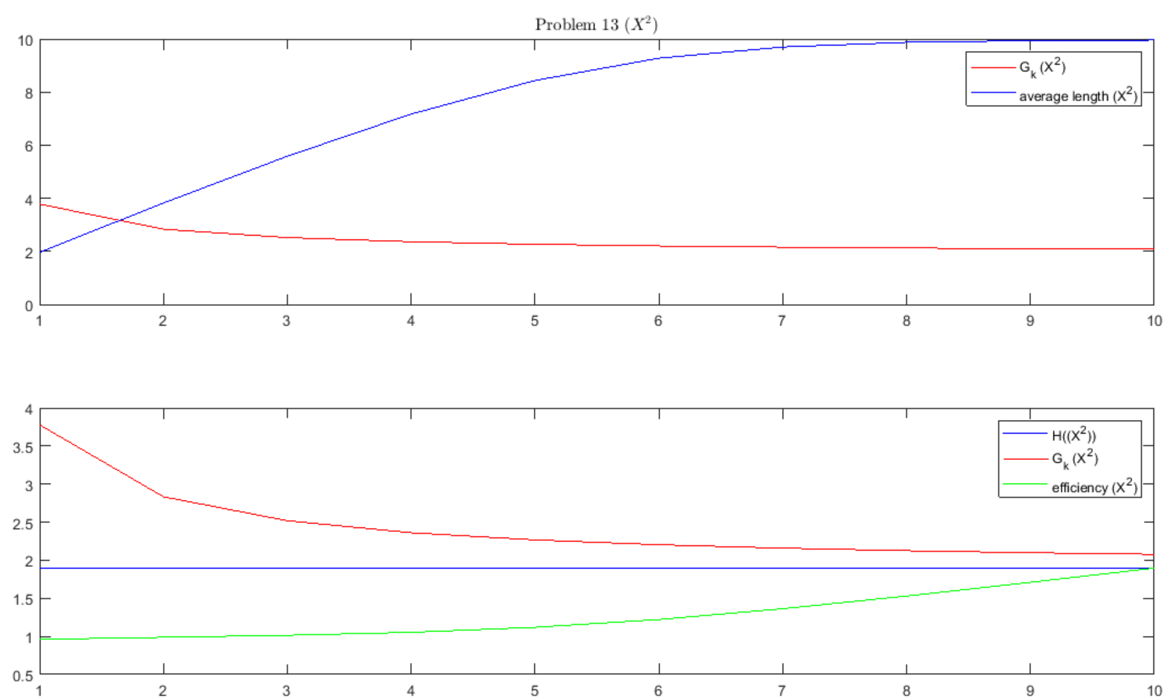
```
Problem 13:
  H(X)  = 0.944544
  H(X2)  = 1.889087
  H(X3)  = 2.833631
```

تصویر ۴ : آنتروپی منابع $X^1$ ، $X^2$ و $X^3$

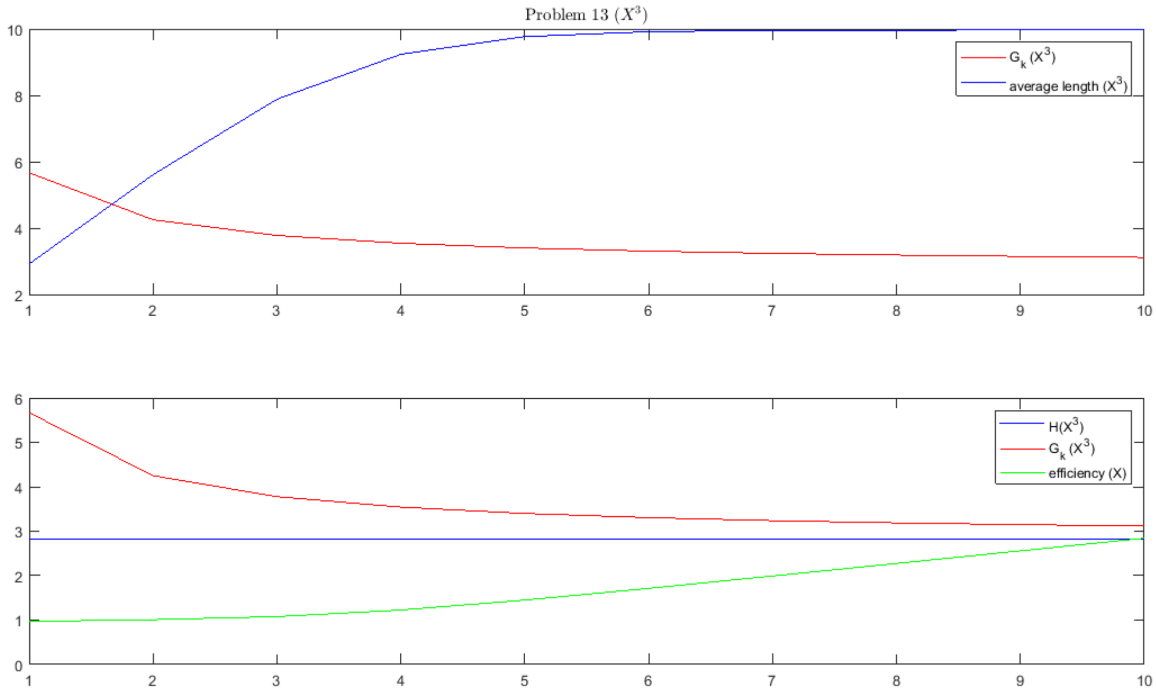
تصاویر ۵ ، ۶ و ۷ به ترتیب نمودار های مربوط به منابع $X^1$ ، $X^2$ و $X^3$ را نشان میدهد. در سوال ۱۴ به تحلیل دقیق این سه تصویر خواهیم پرداخت.




تصویر ۴ : منبع $X^1$




تصویر ۵ : منبع $X^2$




تصویر ۶ : منبع $X^3$

## سوال ۱۴ : تحلیل نمودار های دو قسمت قبل

در این بخش ابتدا به بررسی این مشکل که چرا بهره کدینگ بزرگتر از ۱ شده می‌پردازیم. سپس فرض می‌کنیم حالت ایده آل هستیم و تحلیل ها را انجام می‌دهیم.

### - چرا بهره کدینگ بزرگتر از ۱ شده است؟

برای منبع اول بی حافظه $X^1$ با توجه به اینکه $k = 10$ و ۳ نوع حرف مختلف داریم، یعنی سمبل های یکتا $3^{10}$ هستند و باید یک رشته طولانی در حد ۱۰۰ میلیون حرف داشته باشیم تا مطمئن باشیم احتمال هر سمبل را برای محاسبه متوسط طول کد هافمن به درستی حساب کرده ایم. اما با توجه به کندی تولید رشته این مورد خیلی زمانبر است و چون با طول رشته ۱۰۰۰ شبیه سازی صورت گرفته است، طول متوسط به درستی حساب نشده است. این موضوع در منابع $X^2$ و $X^3$ ، بیشتر اثر می‌کند. برای $k = 10$ ، منبع $X^2$،$9^{10}$ سمبل یکتا دارد و برای منبع $X^3$،$27^{10}$ سمبل یکتا دارد. به همین منظور صرفا برای $k$ های کوچکتر تحلیل را انجام می‌دهیم.

### - تحلیل بخش های ۱۲ و ۱۳:

۱. در منبع با حافظه با افزایش $k$،$G_k$ به آنتروپی منبع میل می‌کند.

۲. در منابع بی حافظه $G_k$ ثابت است. زیرا تولید هر سمبل به سمبل قبل بستگی ندارد و طول کلمات اهمیتی ندارد. (البته چون در این پروژه $G_k$ بصورت تقریبی محاسبه شد، این موضوع خیلی واضح نیست.)

۳. در منابع بی حافظه بهره کدینگ ثابت است. زیرا اضافات ناشی از احتمالات شرطی وجود ندارد و تولید هر سمبل مستقل است.

۴. در منبع با حافظه با افزایش طول کلمات ، بهره کد افزایش یافته تا به ۱ میل کند. البته باید توجه کرد که با افزایش طول کلمات، پهنای باند مورد نیاز نیز افزایش می‌یابد و نمیتوان از حدی بیشتر طول کلمات سمبل ها را افزایش داد.

۵. در منبع بی حافظه برای اینکه بهره کد به ۱ میل کند، باید متوسط طول کد هافمن به صورت خطی زیاد شود. یعنی نسبت متوسط طول کد هافمن به $k$ ثابت است.

۶. بهره کد برای منابع بی حافظه $X^2$ و $X^3$ از $X^1$ بزرگتر است. زیرا طول سمبل ها افزایش می‌یابد.

۷. بهره کدینگ و آنتروپی منابع بی حافظه از با حافظه بزرگتر است؛ زیرا ابهام در منابع بی حافظه کمتر است. چون آنتروپی منبع با حافظه شرطی است و آنتروپی شرطی ، ابهام را کاهش میدهد.