פתרון תרגיל 1

שאלה 1:

ראינו שהמעריך $\hat{n}_U(x) = 2^x - 1$, כאשר x הוא מונה מוריס, הוא מעריך בלתי מוטה של הספירה האמיתית של n.

א. צ"ל את ה-MLE, $\hat{n}_{MLE}(x)$, כאשר ערך ה-Morris Counter הוא $x = 1, \dots, 10$.

$$\hat{n}_{MLE}(X) = \underset{n}{\operatorname{argmax}} f(X{:}\,n)$$

צ"ל למצא עבור כל x_i את הערך של n, עבורו הסיכוי לקבל $x = x_i$ הוא הגבוה ביותר.

הערך של $f(x;n)$ תלוי במצב הקודם שהיה, או שהאיבר החדש גרם להעלאה של x או שלא.

הסתברות המקרה שבו העלנו את x היא $2^{-(x-1)}$, וזו כופלת את ההסתברות של המצב הקודם, כלומר, $f(x-1;n-1)$. באותו אופן, הסתברות המקרה שלא העלנו את x היא $(1-2^{-x})$ והיא כופלת את $f(x;n-1)$. קיבלנו את הנוסחה הרקורסיבית הבאה:

$$f(x;n) = (1-2^{-x})f(x;n-1) + 2^{-(x-1)}f(x-1;n-1)$$

נשים לב שבנוסף אנחנו יודעים את תנאי ההתחלה (עבור האיבר הראשון הסיכוי להגיע ל-$x = 1$ הוא בדיוק 1).

$$f(x = 1, n = 1) = 1$$

בנוסף, נשים לב לאבחנה שלא ייתכן מצב שבו הגענו ל-x יותר גבוה מ-n (אנחנו מגדילים את הערך של x פעם אחת לכל היותר עבור כל n והם מתחילים מאותו ערך...).

$$f(x > n, n) = 0$$

<u>אני מניח שהחלק הבא לא היה נדרש והוא מובא פה בקצרה לשם שלמות התשובה:</u>

יהי x כלשהו, עבור כל $n \geq \hat{n}_{MLE}$, נוכיח כי $f(x;n) \geq f(x;n+1)$ כלומר $\Delta(x,n) = f(x;n) - f(x;n+1) \geq 0$, ולכן נקבל כי בפעם הראשונה שהגענו לירידה בערכים עבור x מסוים נוכל להפסיק לבדוק (מאחר וזהו ה-MLE הדרוש).

נוכיח באינדוקציה על x: עבור $x = 1$ נקבל שהפונקציה מצטמצמת ל- $f(x=1,n) = \frac{1}{2^{n-1}}$ ולכן ברור כי היא יורדת החל מ-1 $= \hat{n}_{MLE}$. נניח כי הטענה נכונה לכל $x < x'$ ונוכיח אותה עבור $x = x'$. נניח כי עבור x', קיים ה- $\hat{n}_{MLE}$ המתאים לו. ונוכיח באינדוקציה על n:

נשים לב כי מאחר וקיים ערך $\hat{n}_{MLE}$ שהוא מקסימאלי, עבור כל $n \neq \hat{n}_{MLE}$ נקבל כי $f(x,n) \leq f(x,\hat{n}_{MLE})$ ובפרט עבור $n = \hat{n}_{MLE} + 1$. נניח כי הטענה נכונה לכל $k < n$ ונוכיח אותה עבור n.

$$\begin{cases} f(x;n) = (1-2^{-x})f(x;n-1) + 2^{-(x-1)}f(x-1;n-1) \\ f(x;n+1) = (1-2^{-x})f(x;n) + 2^{-(x-1)}f(x-1;n) \end{cases}$$

$$\begin{aligned}\Rightarrow \Delta(x,n) &= f(x;n+1) - f(x;n)\\ &= (1-2^{-x})\big(f(x;n) - f(x;n-1)\big)\\ &+ 2^{-(x-1)}\big(f(x-1;n) - f(x-1;n-1)\big)\\ &= (1-2^{-x})\Delta(x,n-1) + 2^{-(x-1)}\Delta(x-1,n-1)\end{aligned}$$

ע"פ הנחת האינדוקציה (על n) נקבל כי החלק הראשון $\Delta(x,n-1)\le 0$ וכן מאחר ו-$\hat{n}_{MLE}$ היא פונקציה לא יורדת, ע"פ הנחת האינדוקציה (על x) נקבל כי $\Delta(x-1,n-1)\le 0$ ולכן:

$$\Delta(x,n) = f(x;n+1) - f(x;n) \le 0$$

🡨 מצאנו כי עבור x מסוים ניתן לחשב את הערכים של $f(x,n)$ עד הערך הראשון שלא עולה על קודמו, וזהו ה-$\hat{n}_{MLE}$.

נחשב באמצעות תכנות דינאמי את המטריצה הנ"ל ונמצא את המקסימום עבור כל x.

התוצאות:

x	$\hat{n}_{MLE}$	Max-Likelihood
1	1	1
2	3	0.625
3	8	0.523670197
4	19	0.486695161
5	39	0.470369477
6	80	0.462512782
7	162	0.45869213
8	325	0.456812012
9	652	0.455876987
10	1306	0.45541105

הקוד מצורף בסוף התשובות (נספח א') – כולל את סעיפים א' ו-ב' של שאלה 1.

ב. נחשב את הערכים הבאים:

(a) $Bias(\hat{n}_{MLE},n) = E[\hat{n}_{MLE}] - n$

(b) $MSE(\hat{n}_{MLE},n) = E[(\hat{n}_{MLE}-n)^2]$

$$= E[\hat{n}_{MLE}^2 - 2\cdot\hat{n}_{MLE}\cdot n + n^2] = E[\hat{n}_{MLE}^2] - 2n\cdot E[\hat{n}_{MLE}] + n^2$$

(c) $NRMSE(\hat{n}_{MLE},n) = \frac{\sqrt{MSE(\hat{n}_{MLE})}}{n}$

(d) $CV[\hat{n}] = \frac{\sigma}{\mu} \quad ;\sigma = \sqrt{V[\hat{n}]}$

ע"פ מה שראינו בשיעור, $V[\hat{n}] = \frac{3}{2}n^2 + \frac{3}{2}n + 1 - (n+1)^2 = \frac{1}{2}n^2 - \frac{1}{2}n$

$$\Rightarrow \sigma = \sqrt{\frac{1}{2}(n^2-n)}$$

וכן, עבור מעריך בלתי מוטה $\mu = E[\hat{n}] = n$ ונקבל:

$$CV[\hat{n}] = \sqrt{\frac{n^2-n}{2n^2}} = \sqrt{\frac{n-1}{2n}}$$

n	$Bias(\hat{n}_{MLE}, n)$	$MSE(\hat{n}_{MLE})$	$NRMSE(\hat{n}_{MLE})$	$CV[\hat{n}]$
1	0.0	0.0	0.0	0.0
2	0.0	1.0	0.5	0.5
3	0.125	4.125	0.677003200386	0.57735026919
4	0.328125	9.484375	0.769917812172	0.612372435696
5	0.580078125	16.966796875	0.823815437462	0.632455532034
6	0.862091064453	26.4124450684	0.856550657975	0.645497224368
7	1.16192817688	37.68080616	0.876924195174	0.654653670708
8	1.47153985128	50.6706994362	0.889791929999	0.661437827766
9	1.78563172738	65.320132343	0.898009680878	0.666666666667
10	2.10075975438	81.5989877533	0.903321580354	0.67082039325

ג. נוכיח כי $MSE[\hat{\theta}] = Var[\hat{\theta}] + Bias(\hat{\theta}, \theta)^2$ לכל מעריך $\hat{\theta}$ של θ.

אנו יודעים כי:

$$Bias(\hat{\theta}, \theta) = E[\hat{\theta}] - \theta \Rightarrow Bias(\hat{\theta}, \theta)^2 = (E[\hat{\theta}])^2 - 2 \cdot \theta \cdot E[\hat{\theta}] + \theta^2$$

$$Var[\hat{\theta}] = E\left[(\hat{\theta} - E[\hat{\theta}])^2\right] = E[\hat{\theta}^2] - (E[\hat{\theta}])^2$$

$$MSE(\hat{\theta}, \theta) = E\left[(\hat{\theta} - \theta)^2\right] = E[\hat{\theta}^2 - 2 \cdot \theta \cdot \hat{\theta} + \theta^2] = E[\hat{\theta}^2] - 2 \cdot \theta \cdot E[\hat{\theta}] + \theta^2$$

ולכן:

$$Var[\hat{\theta}] + Bias(\hat{\theta}, \theta)^2 = E[\hat{\theta}^2] - (E[\hat{\theta}])^2 + (E[\hat{\theta}])^2 - 2 \cdot \theta \cdot E[\hat{\theta}] + \theta^2$$
$$= E[\hat{\theta}^2] - 2 \cdot \theta \cdot E[\hat{\theta}] + \theta^2 = MSE(\hat{\theta}, \theta)$$

$$\Rightarrow Var[\hat{\theta}] + Bias(\hat{\theta}, \theta)^2 = MSE(\hat{\theta}, \theta) \blacksquare$$

ד. נגדיר $A = \frac{1}{k}\sum_{i=1}^{k} \hat{n}_{MLE}(x_i)$ ו- $B = \frac{1}{k}\sum_{i=1}^{k} \hat{n}_U(x_i)$. צ"ל: $NRMSE(A)$ ו- $NRMSE(B)$.

ע"פ מה שלמדנו בשיעור הראשון, כאשר לוקחים מעריך שהוא ממוצע של k מעריכים בלתי תלויים התוחלת נשארת זהה וה-Variance יורד פי k, ולכן ה-CV יורד פי $\sqrt{k}$.

כאמור בסעיף ב', $CV[\hat{n}] = NRMSE(\hat{n})$ ולכן נקבל כי: $NRMSE(B) = CV(B) = \frac{CV(\hat{n})}{\sqrt{k}}$

ועבור $n = 8$ ← $NRMSE(B) = \frac{0.6614}{\sqrt{k}}$.

לפי סעיף ג' אנו יודעים כי $Var[A] = MSE(A, n) - Bias(A, n)^2$,

בנוסף $Bias(A, n) = E[A] - n = E[\hat{n}_{MLE}] - n = Bias(\hat{n}_{MLE}, n)$.

$$\Rightarrow NRMSE(A) = \frac{\sqrt{MSE(A)}}{n} = \frac{\sqrt{Var[A] + Bias(A, n)^2}}{n} = \frac{\sqrt{\frac{1}{k} Var[\hat{n}_{MLE}] + Bias(\hat{n}_{MLE}, n)^2}}{n}$$
$$= \frac{\sqrt{\frac{1}{k}(MSE(\hat{n}_{MLE}, n) - Bias(\hat{n}_{MLE}, n)^2) + Bias(\hat{n}_{MLE}, n)^2}}{n}$$

ועבור $n = 8$ ← $NRMSE(A) = \sqrt{\frac{1}{k} \cdot 0.7579 + 0.0338}$.

שאלה 2:

א. נוכיח כי בכל שלב בעיבוד ה-stream נקבל כי $\hat{c}_i \leq c_i$.

נסמן ב-c_i^k וב-$\hat{c}_i^k$ את הערכים אותם יחזירו האלגוריתמים, לאחר עיבוד של k האיברים הראשונים.

נוכיח באמצעות אינדוקציה על n (מספר האיברים אותם עיבדנו) כי לאחר עיבוד כל איבר נקבל $\hat{c}_i^n \leq c_i^n$ (ולכן נקבל כי באופן כללי $\hat{c}_i \leq c_i$).

עבור המצב ההתחלתי $n = 0$: הערך c_i^0 ע"פ הגדרתו הוא 0, וכן, מאחר ולא קיים *counter* האלגוריתם $\pm MG$ יחזיר $\hat{c}_i^0 = 0$.

נניח כי הטענה נכונה עבור $n = k$, כלומר לאחר עיבוד של k איברים מה-*stream* קיבלנו כי:

$$\hat{c}_i^k \leq c_i^k$$

וכעת, נוכיח אותה עבור $n = k + 1$.

בהינתן שקיבלנו (j, Δ), נחלק את העולם לשני מצבים:

__אם $j \neq i$ אזי__, הערך האמיתי של c_i לא ישתנה ונקבל כי $c_i^{k+1} = c_i^k$, וכן עבור הערך המקורב:

- אם קיים counter ל-j אזי בהתאם ל-Δ נעלה / נוריד אותו באחד (והדבר לא ישפיע על $\hat{c}_i$) ולכן נקבל $\hat{c}_i^{k+1} = \hat{c}_i^k$.
- אם לא קיים counter ל-j אזי:
  1. אם $\Delta = -1$ לא נבצע כלום ולכן נקבל $\hat{c}_i^{k+1} = \hat{c}_i^k$.
  2. אחרת אם $\Delta = +1$, נוריד את כל ה-counterים הקיימים ב-1 (אם $\hat{c}_i^k = 0$ אזי הוא יישאר 0) ונקבל כי $\hat{c}_i^{k+1} \leq \hat{c}_i^k$.

ובאופן כללי, נקבל $\hat{c}_i^{k+1} \leq \hat{c}_i^k$.

$$\Rightarrow \hat{c}_i^{k+1} \leq \hat{c}_i^k \leq c_i^k = c_i^{k+1}$$

$$\Rightarrow \hat{c}_i^{k+1} \leq c_i^{k+1}$$

__אחרת, $j = i$ אזי__, הערך האמיתי של c_i ישתנה ונקבל $c_i^{k+1} = \max\{0, c_i^k + \Delta\}$, וכן עבור הערך המקורב:

- אם קיים counter ל-i אזי בהתאם ל-Δ נעלה / נוריד אותו ונקבל כי $\hat{c}_i^{k+1} = \hat{c}_i^k + \Delta$.
  - ולכן נקבל כי (מאחר ו-$0 < \hat{c}_i^k \leq c_i^k$ אזי $c_i^{k+1} = c_i^k + \Delta$)

$$\Rightarrow \hat{c}_i^{k+1} = \hat{c}_i^k + \Delta \leq c_i^k + \Delta = c_i^{k+1}$$

$$\Rightarrow \hat{c}_i^{k+1} \leq c_i^{k+1}$$

- אם לא היה קיים counter ל-i אזי הערך המקורב $\hat{c}_i$ יישאר זהה ולכן נקבל $\hat{c}_i^{k+1} = \hat{c}_i^k = 0$.
  נפצל שוב את העולם לשני מקרים:
  אם $\Delta = +1$ אזי:

$$\Rightarrow \hat{c}_i^{k+1} = \hat{c}_i^k \leq c_i^k < c_i^k + 1 = c_i^{k+1}$$

$$\Rightarrow \hat{c}_i^{k+1} \leq c_i^{k+1}$$

אחרת, אם $\Delta = -1$ אזי: הערך האמיתי יהיה $c_i^{k+1} = \max\{0, c_i^k - 1\} \geq 0$ וכן הערך המקורב כאמור יהיה $\hat{c}_i^{k+1} = 0$ ולכן:

$$\Rightarrow \hat{c}_i^{k+1} = 0 \leq c_i^{k+1}$$

קיבלנו כי בכל המקרים לעיל $\hat{c}_i^{k+1} \leq c_i^{k+1}$ ולכן, ע"פ הנחת האינדוקציה המתמטית הטענה נכונה לכל k טבעי, ובפרט ל-$k = n$, כלומר:

$$\Rightarrow \hat{c}_i^n \leq c_i^n$$

$$\Rightarrow \hat{c}_i \leq c_i \blacksquare$$

ב. צ"ל את החסם ההדוק העליון של $c_i - \hat{c}_i$.

נשים לב כי הפער יגדל כאשר נקבל איבר (j, Δ) כך ש-$\hat{c}_j$ לא נמצא וכן $\Delta = +1$, ובמהלך זה נוריד סה"כ k מהסכום הכולל ונתעלם מאיבר נוסף. נתעלם במקרה זה מסה"כ $k + 1$ הופעות בלתי ספורות, נגדיר אותו בתור decrement ונסמן את מספר הפעמים שהוא קורה ב-d. בנוסף, הפער יקטן כאשר נקבל (i, Δ) כש- $\hat{c}_i$ לא נמצא וכן $\Delta = -1$. בכל שאר הצעדים הפער לא ישתנה.

ההפרש המרבי עבור מונה בודד: $c_i - \hat{c}_i \leq d$.

נסמן כעת: $m = \sum_i c_i$ וכן, $m' = \sum_i \hat{c}_i$. ונגדיר שני משתני עזר: Δ_+ - מספר הפעמים בהם קיבלנו איבר עם $\Delta = +1$, וכן Δ_- - מספר הפעמים בהם קיבלנו איבר עם $\Delta = -1$ והוא אכן הוריד את הערך של המונה האמיתי c_i ב-1 (כלומר לפניו התקיים $c_i > 0$).

$$\Rightarrow m = \Delta_+ - \Delta_-$$

נסמן ב-$\mathbb{d}$ את מספר האיברים שבהם $\Delta = -1$ וכתוצאה ממעבר עליהם <u>השתנה</u> הערך של $\hat{c}$.

$$\Rightarrow m - m' = d(k + 1) - (\Delta_- - \mathbb{d})$$

$$\Rightarrow \Delta_+ - \Delta_- - m' = d(k + 1) - (\Delta_- - \mathbb{d})$$

$$\Rightarrow \Delta_+ - m' = d(k + 1) + \mathbb{d}$$

$$\Rightarrow d = \frac{\Delta_+ - m' - \mathbb{d}}{k + 1}$$

$$\Rightarrow c_i - \hat{c}_i \leq d = \frac{\Delta_+ - m' - \mathbb{d}}{k + 1}$$

אנחנו כמובן יכולים לשמור במהלך המעבר את שני המונים (אחד עבור Δ_+ והשני עבור $\mathbb{d}$) ולסכום את כל המונים $\hat{c}_i$ כדי להגיע ל-m'.

במידה ולא קיבלנו איבר עם $\Delta = -1$ קיבלנו רדוקציה של הבעיה והאלגוריתם ל- MG והוכחנו בכיתה שבמקרה זה החסם הינו הדוק ($\Delta_+ = m, \mathbb{d} = 0$ 🡐). וכן, ניתן לחשוב על מקרה בו כל פעם שמבצעים decrement מונה בודד $\hat{c}_x$ בטוח מושפע, וב-stream לא קיימים איברים בעלי $\Delta = -1$ (או שהם תמיד השפיעו גם על $\hat{c}$) אזי במקרה זה אכן נקבל $c_x - \hat{c}_x = d$ ולכן החסם הדוק.

שאלה 3:

א. צ"ל מעריך בלתי מוטה $\hat{c}$ לגודל c (מספר הפקטות) ב-flow.

המעריך $\hat{c}_i$ עבור c_i (המתאים ל-flow i), יחזיר במידה וקיים מונה בעל הערך x עבור ה-flow את $\frac{x}{p}$ ואם לא קיים יחזיר 0.

נראה כי $E[\hat{c}] = c$: נסתכל על flow בודד, ונשים לב שההסתברות לדגום פקטה היא p ואינה תלויה בקלט. זוהי התפלגות בינומית ולכן $x \sim B(c_i, p)$.

$$\Rightarrow E[\hat{c}_i] = E\left[\frac{x}{p}\right] = \frac{c_i p}{p} = c_i$$

ב. צ"ל את ה-$Var[\hat{c}]$.

שוב נסתכל על flow בודד:

$$Var[\hat{c}_i] = Var\left[\frac{x}{p}\right] = \frac{1}{p^2} \cdot Var[x] = \frac{1}{p^2} \cdot \big(c_i p(1-p)\big) = c_i \cdot \frac{1-p}{p}$$

ג. צ"ל את $CV[\hat{c}]$.

$$CV[\hat{c}] = \frac{\sqrt{V[\hat{c}]}}{E[\hat{c}]} = \sqrt{c_i \cdot \frac{1-p}{p}\frac{1}{c_i^2}} = \sqrt{\frac{1-p}{c_i p}}$$

ד. צ"ל שסיכומי *NetFlow* ניתנים לאיחוד.

נניח כי ישנם שני *stream*ים, כאשר בהתבוננות על *flow* בודד ה-*stream* הראשון מחזיר $\hat{c}_1$ עבור c_1 פקטות ב-*flow* הנ"ל והשני מחזיר $\hat{c}_2$ עבור c_2 פקטות ב-*flow* הנ"ל בהתאמה.

עבור flow בודד נגדיר את המיזוג כך (כאשר כמובן אין צורך להוסיף מונים אם בשני ה-streamים לא נוצר מונה עבור flow מסוים):

$$\hat{c} = \hat{c}_1 + \hat{c}_2$$

נשים לב שבמקרה זה, עבור stream מחובר רציף היינו מקבלים (ע"פ סעיף א'):

$$E[\hat{c}] = c_1 + c_2$$

וכן עבור איחוד של שני ה-streamים ($\hat{c}_1$ ו-$\hat{c}_2$ בלתי תלויים):

$$E[\hat{c}] = E[\hat{c}_1] + E[\hat{c}_2] = c_1 + c_2 \blacksquare$$

ה. בהתחשב ב-*stream* בעל m פקטות, צ"ל איך לחלק את הפקטות ל-*flow*ים בכדי למקסם את כמות המונים.

נסתכל על *flow* בודד, ונמצא מה ההסתברות לפתוח *counter* חדש כפונקציה של מיקום הפקטה ב-*flow* (הסיכוי שכן דגמנו אותה אבל לא דגמנו את כל הקודמות):

$$\Pr[new\ counter\ on\ packet\ i] = p \cdot (1-p)^{i-1}$$

וכן אם נסכום על כל הפקטות ב-*stream* אזי מספר המונים יהיה (עבור k *flow*ים):

$$n = \sum_{j=1}^{k} \sum_{i=1}^{c_k} \Pr[new\ counter\ on\ packet\ i] = \sum_{j=1}^{k} \sum_{i=1}^{c_k} p \cdot (1-p)^{i-1}$$

ברור כי ההסתברות עבור כל פקטה לפתוח *counter* חדש יורדת עם מיקומה ב-*stream* ולכן נעדיף "לנצל" אותה כאשר היא במקסימום, כלומר, אם נחלק כל פקטה ל-*flow* שונה אזי נקבל כי ההסתברות לפתוח *counter* חדש עבור כל פקטה היא בדיוק p, ולכן נקבל את המקסימום של n.

נקבל כי:

$$n = \sum_{j=1}^{k} \sum_{i=1}^{c_k} p \cdot (1-p)^{i-1} = \sum_{j=1}^{c} \sum_{i=1}^{1} p \cdot (1-p)^{i-1} = \sum_{j=1}^{c} p = cp \blacksquare$$

נספח א' (קוד python לשאלה מספר 1 סעיפים א' ו-ב'):

```
MAX_X = 10 # the maximum value for x
MAX_N = 10 # the maximum value for n (for computing section B)
# Using dynamic programing we compute each value of f(x,n) once and store it inside this table:
table = [[] for x in xrange(MAX_X)]
# a row in the table holds all values of possibilities for where x is the (row-index + 1)
# the index inside the row corresponds to value of n.
def f(x, n):
    # This function retrives the value if previously computed, else computes it first.
    # f(x;n) = table[x-1][n-1]
    if len(table[x-1]) < n:
        fill_row(x-1, n)
    return table[x-1][n-1]

def compute(x, n):
    # This function computes the value of f(x,n).
    if x == 1 and n == 1:
        f_xn = 1.0
    elif x > n:
        f_xn = 0.0
    else:
        f_xn = (1 - 2**(-x)) * f(x, n - 1)
        if x > 1:
            f_xn += 2**(-(x - 1)) * f(x - 1, n - 1)
    return f_xn

def fill_row(i, max_n = -1):
    # This function fills the row (which corresponds to a value of x) until we either
    # reach the maximum likelihood for that value of x or until we reach max_n (inorder
    # to calculate another value: f(x, max_n) which is needed)
    x = i + 1                   # the value of x
    last = 0                    # saves the last value
    value = last                # saves the current value
    n = len(table[i]) + 1       # the start n (the first unyet calculated value)
    while not (last > value or ((max_n != -1) and (n > max_n))):
        last = value            # update the last value
        value = compute(x, n)   # get the computed value for f(x, n)
        table[i].append(value)  # save the value in the table
        n = n + 1               # increase the value of n

def fill_table(max_x = MAX_X):
    # This function fills the entire table up to the value of x = max_x
    for i in xrange(max_x):
        fill_row(i)

def print_sum():
    # prints the outcome of the run
    print "x\tn_mle\tMax-Likelihood"
    print "-" * 30
    n_mle = []
    for i in xrange(MAX_X):
        max_value = max(table[i])
        n_value = table[i].index(max(table[i])) + 1
        print i+1, "\t", n_value, "\t", max_value
        n_mle.append(n_value)
    print ""

    print "n\ta\tb\tc\td"
    print "-" * 40
    for j in xrange(MAX_N):
        n = j + 1
        col = [f(i+1, n) for i in xrange(n)]
        E_mle =  sum([n_mle[i]      * p for i, p in enumerate(col)])
        E_mle2 = sum([n_mle[i] ** 2 * p for i, p in enumerate(col)])
        a = E_mle - n
        b = E_mle2 - 2 * n * E_mle + n ** 2
        c = b ** 0.5 / n
        d = ((n - 1) / (2.0 * n)) ** 0.5
        print n, "\t", a, "\t", b, "\t", c, "\t", d
    print ""

def main():
    fill_table()
    print_sum()
if __name__ == "__main__":
    main()
    import os
    os.system("pause")
```