שיטות נומריות: תרגול 1

מייל:pirvandy@gmail.com

אני רוצה להתחיל את הקורס עם ציטוט של בנימין ד'יזראלי (ראש ממשלת בריטניה):

"ישנם שלושה סוגי שקרים – שקרים, שקרים ארורים וסטטיסטיקה".

אני הייתי מרחיב את היריעה ואומר כי ישנם 4 שקרים:

1. שקר לבן.
2. שקר.
3. שקר ארור.
4. סטטיסטיקה

שיטות נומריות הן השקרים הלבנים של המתמטיקה. לא פעם אנו נתקלים בבעיה שאין לה פתרון אנליטי, בבעיות אלו מדי פעם אנו מכופפים את חוקי המתמטיקה ומפעילים מניפולציות. מניפולציות אלה הן שיטות נומריות להשגת תשובה מספיק קרובה לאמת אבל היא שונה ממש מן האמת. נביא דוגמא לכך על ידי משוואה שכולנו יכולים לפתור:

המרת בסיסים וייצוג מספרים בבסיסים שונים:

הצגה של מספר בבסיס 2:

נתון מספר X . מחפשים סדרה $\{a_n\}$ כך ש:

$$X = a_n * 2^n + a_{n-1} * 2^{n-1} + \ldots + a_0 * 2^0 + a_{-1} * 2^{-1} + \ldots$$

כאשר: $a_n \in \{0,1\}$, ואז ההצגה של $a_n, a_{n-1}, \ldots, a_0, \ldots, a_{-4}, \ldots$ היא ההצגה בבסיס 2 של X .

דוגמא:

X=53

אפשרות ראשונה לפתרון, לפי החזקה הגדולה ביותר

שיטה נוספת:

$$35 = a_n * 2 + \ldots + a_0 * 2^0 + \ldots + a_{-5} * 2^{-5}$$

מעבר מבסיס 2 לבסיס 10:

נתונה הסדרה $a_n, a_{n-1}, \ldots, a_0, \ldots, a_{-4}, \ldots$.

משווים את הסכום:

$$X = a_n * 2^n + a_{n-1} * 2^{n-1} + \ldots + a_0 * 2^0 + a_{-1} * 2^{-1} + \ldots$$

דוגמא: 101010.01_2

$$1*2^5+1*2^3+1*2^1+1*2^{-2}=32+8+2+0.25=42.25$$

בבסיס 8 X מוגדר כך:

$$X = a_n * 8^n + a_{n-1} * 8^{n-1} + \ldots + a_0 * 8^0 + a_{-1} * 8^{-1} + \ldots$$

כאשר $a_n \in \{0,1,2,\ldots 7\}$

בבסיס 16 X מוגדר כך:

$$X = a_n * 16^n + a_{n-1} * 16^{n-1} + \ldots + a_0 * 16^0 + a_{-1} * 16^{-1} + \ldots$$

כאשר $a_n \in \{0,1,\ldots 9, A, B, C, D, E, F\}$

מעבר מבסיס 2 לבסיס 16:

כל אות או ספרה בבסיס 16 שווה ל-4 אותיות בבסיס 2. לדוגמא עבור $(A170F)_{16}$:

$$\underset{(1010)_2}{A}\ \underset{(0001)_2}{1}\ \underset{(0111)_2}{7}\ \underset{(0000)_2}{0}\ \underset{(1111)_2}{F}$$

לסיכום - כדי להמיר מבסיס 16 לבסיס 2 צריך להחליף/להמיר כל תו (מספר או אות) ב-4 אותיות של בסיס 2 בנפרד.

<u>במעבר מבסיס 2 לבסיס 16</u> - מקבצים כל 4 ספרות (מתחילים מימין) וממירים אותם לאות של בסיס 16 בנפרד.

<u>דוגמא</u>: (101110)

$$\underset{(2)_{16}}{0010}\ \underset{(E)_{16}}{1100}=(2)_{16}(E)_{16}=(2E)_{16}$$

דגש

במעבר מבסיס 2 לבסיס 8 - דומה מאוד לנ"ל, במקום להמיר 4 ספרות לספרה אחת של בסיס 16, ממירים 3 ספרות של בסיס 2 לספרה של בסיס 8.

ייצוג מספרים במחשב:

קיימות שתי שיטות עיקריות:

1. Fixed Point - עיגול. מציגים כל מספר על ידי X ספרות אחרי הנקודה העשרונית.

למשל עבור $X=2$:

א. התצוגה עבור π תהיה: 3.14

ב. התצוגה עבור $\frac{1}{8}=0.125$ תהיה: 0.12

ג. התצוגה עבור 0.005 תהיה 0 .

2. Floating Point - ספרות משמעותיות. פורמט זה מציג כל מספר עד כדי X ספרות משמעותיות, כאשר אפשר להוסיף 0 לפני ואחרי המספר ככל שנרצה.

למשל עבור $X=2$:

א. התצוגה עבור π תהיה: 3.1

ב. התצוגה עבור 0.125 תהיה: 0.12

ג. התצוגה עבור 0.0054 תהיה 0.0054

ד. התצוגה עבור 1000.01 תהיה 1000 (מעגל לקרוב ביותר)

ה. התצוגה עבור 1234 תהיה 1200

ו. התצוגה עבור 1274 תהיה 1300

ז. התצוגה עבור 0.127 תהיה 0.13

<u>באופן כללי:</u> ייצוג בנקודה צפה מחולק ל-3 רכיבים:

$$X = \sigma * (m)_B * B^{\exp}$$

פירוש מרכיבי ההצגה σ הוא סימן $(sign)$ של המספר, קרי חיובי או שלילי.
$(m)_B$ המנטיסה היא מספר הספרות המשמעותיות בבסיס B , נזכור כי במחשב הבסיס הוא בינארי. בעוד ה- $(\exp)$ הוא המיקום של הנקודה העשרונית במספר.

<u>**דוגמא**</u>: נניח לעת עתה שהבסיס הוא עשרוני, קרי $B=10$:ΔΔ

$$12 = 1 * (1.2) * 10^1$$

$$1.2 = 1 * (1.2) * 10^{-1}$$

<u>דוגמא</u>: נניח לעת עתה שהבסיס הוא עשרוני, קרי $B=10$ $\triangle\triangle$:

$$12=1*(1.2)*10^{1}$$

$$1.2=1*(1.2)*10^{-1}$$

בדוגמא זו אנו רואים כי הסימן הוא חיובי, המנטיסה זהה, יש רק שינוי בסקאלה של האקספוננט $(\exp)$.

$$-1.2=-1*(1.2)*10^{-1}$$

$$1234.567\ldots=1*(1.234567\ldots)*10^{3}$$

במחשב ישנם 2 תקנים להצגת מספרים בשיטת נקודה צפה: $float$ ו- $double$, שניהם מיוצגים בבסיס בינארי $(0,1)$.

בתקן $float$ משתמשים ב-32 סיביות, כאשר כל סיבית מקבלת ערך 0 או 1.

המספר בנוי בצורה הבאה:

σ	$\exp$	$(m)_B$
1	8	23

כלומר:

1. $sign$ - סיבית 1, עבור חיובי מקבלת את הערך 0, אחרת 1.
2. $\exp$ - מיקום הנקודה העשרונית. יש 8 סיביות המספר הגדול ביותר הוא 111,111,11 זה שווה ל-255. אנו רוצים לשמור מספרים בסדר גודל גדול/עצום (אנו רוצים לשמור חזקה גבוהה) וכן רוצים לשמור חזקה נמוכה, לשם כך אנו נוסיף אוטומטית 127 לחזקה. כלומר במקום לשמור 2^{-107} נרשום אותו בתור $2^{-107+127}=2^{20}$, וכך המחשב ישמור את הערך ב- $\exp=00010100$. סדרי הגודל שניתן לשמור הם $[2^{-126},2^{127}]$. נרשום על כך הערה בהמשך.
3. $(m)_B$ – המנטיסה, מיוצגת ע"י 23 סיביות. המנטיסה היא סכום הספרות המשמעותיות במספר.

תקן double נראה כך:

σ	exp	$(m)_B$
1	11	52

תרגיל: ייצגו את המספר 46.75 בתקן $float$ (להעביר לבינארי).

נאחד את מה שקיבלנו עד עכשיו ונקבל:

$$46.75 = 46 + 0.75 = 2^5 + 2^3 + 2^2 + 2^1 + 2^{-1} + 2^{-2}$$

$$= \underbrace{2}_{\text{בסיס}}{}^{\overbrace{5}^{\text{חזקה לפני נרמול הפוך}}} * \underbrace{[1 + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-6} + 2^{-7}]}_{\text{מנטיסה}}$$

וכעת:

$$exp = 5 + 127 = 132 = 128 + 4 = 2^7 + 2^2$$

ולכן בסה"כ:

0	1	0	0	0	0	1	0	0	0	1	1	1	0	1	1	0	.....	0
$sign$	7	6	5	4	3	2	1	0	-1	-2	-3	-4	-5	-6	-7	-8	.....	23-

הערות:

1. החזקה הנמוכה ביותר: 126-, בייצוג בינארי: 00000001
$$\underbrace{00000001}_{1}-127=-126$$
2. החזקה הגבוהה ביותר: 127, בייצוג בינארי: 11111110
$$\underbrace{11111110}_{154}-127=127$$
3. אם החזקה (exp) היא כולה אחדות והמנטיסה גם כולה אחדות, זה מייצג ∞ .
4. כאשר החזקה כולה אפסים והמנטיסה גם כולה אפסים, זה מייצג 0.
5. אם החזקה כולה אחדות ולפחות מקום אחד במנטיסה הוא 0, זה מייצג NaN.