Khảo sát sự thay đổi độ nhảy cóc qua các mốc số nguyên

Xét số float, với việc phân tích cấu trúc ở phần trên và cơ chế lưu trữ khi số bit cần để biểu diễn số nguyên vượt quá số bit của phần định trị (23 + 1), chúng ta dự đoán được các mốc thay đổi nhảy cóc sẽ là 2^n với n >= 24. Ta thực hiện việc khảo sát với số float bằng ngôn ngữ Java như sau:

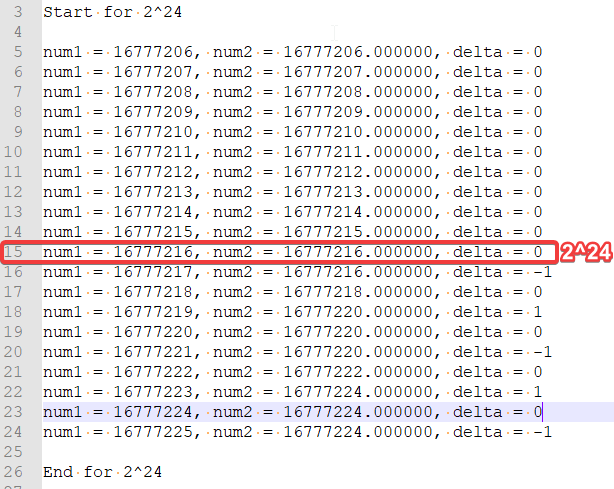
* Cho các số nguyên trong khoảng [2^n – 10, 2^n + 10] (với 24 <= n <= 30)
* Ta biểu diễn các số nguyên đó bằng 2 kiểu dữ liệu: kiểu int và kiểu float để từ đó nhận xét độ chênh lệch giữa 2 cách biểu diễn.

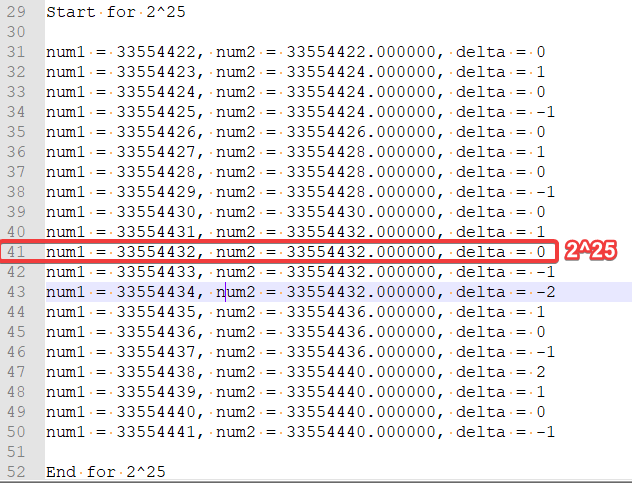
Kết quả và nhận xét:

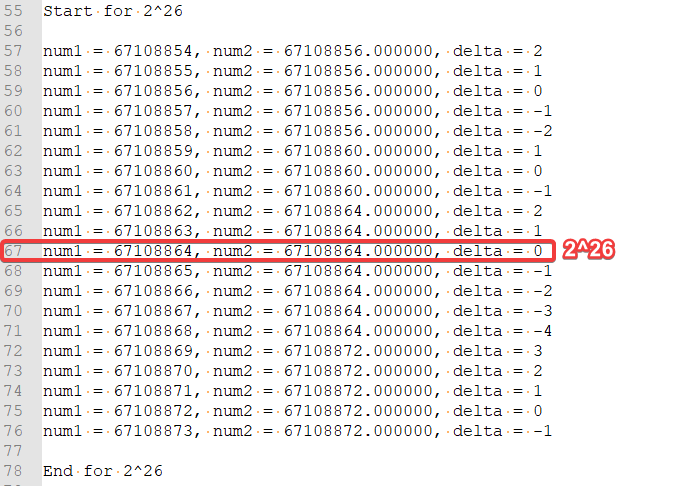
Code snippet và kết quả output được đính kèm trong bài nộp tương ứng ở 2 file: int\_to\_float\_java\_output.txt và int\_to\_float\_java\_code\_snippet.txt

Chú thích:

* num1 là số nguyên biểu diễn bằng kiểu int
* num2 là số nguyên biểu diễn bằng kiểu float
* delta = num2 – num1







Với các số < 2^24, kiểu dữ liệu float có thể biểu diễn chính xác giá trị số nguyên.

Từ 2^24 trở đi, giả sử đang khảo sát các số nguyên lân cận 2^n: nhảy cóc xuất hiện với giá trị tối đa là 2^(n-24) – 1, khoảng cách giữa các bước nhảy là 2^(n-24), nên các giá trị số nguyên i với [i mod 2^(n - 24) != 0] sẽ bị biểu diễn sai

Mốc của sự thay đổi độ nhảy cóc là số nguyên 2^n, qua số nguyên này bước nhảy sẽ biến đổi từ 2^(n-1-24) sang 2^(n-24)

Sự biến thiên độ lệch sẽ có một số bất thường không theo chu kỳ (như với 2^25 độ lệch sẽ biến đổi -1 -2 1 0 -1 2 1 0 -1 …) do cơ chế làm tròn cho số floating point của Java sẽ làm tròn số nguyên đầu vào tới số nguyên mà float có thể biểu diễn được gần nhất

Đồ thị thể hiện sự thay đổi độ nhảy cóc qua các mức số nguyên