NASA HW1 References

B11901164 陳秉緯

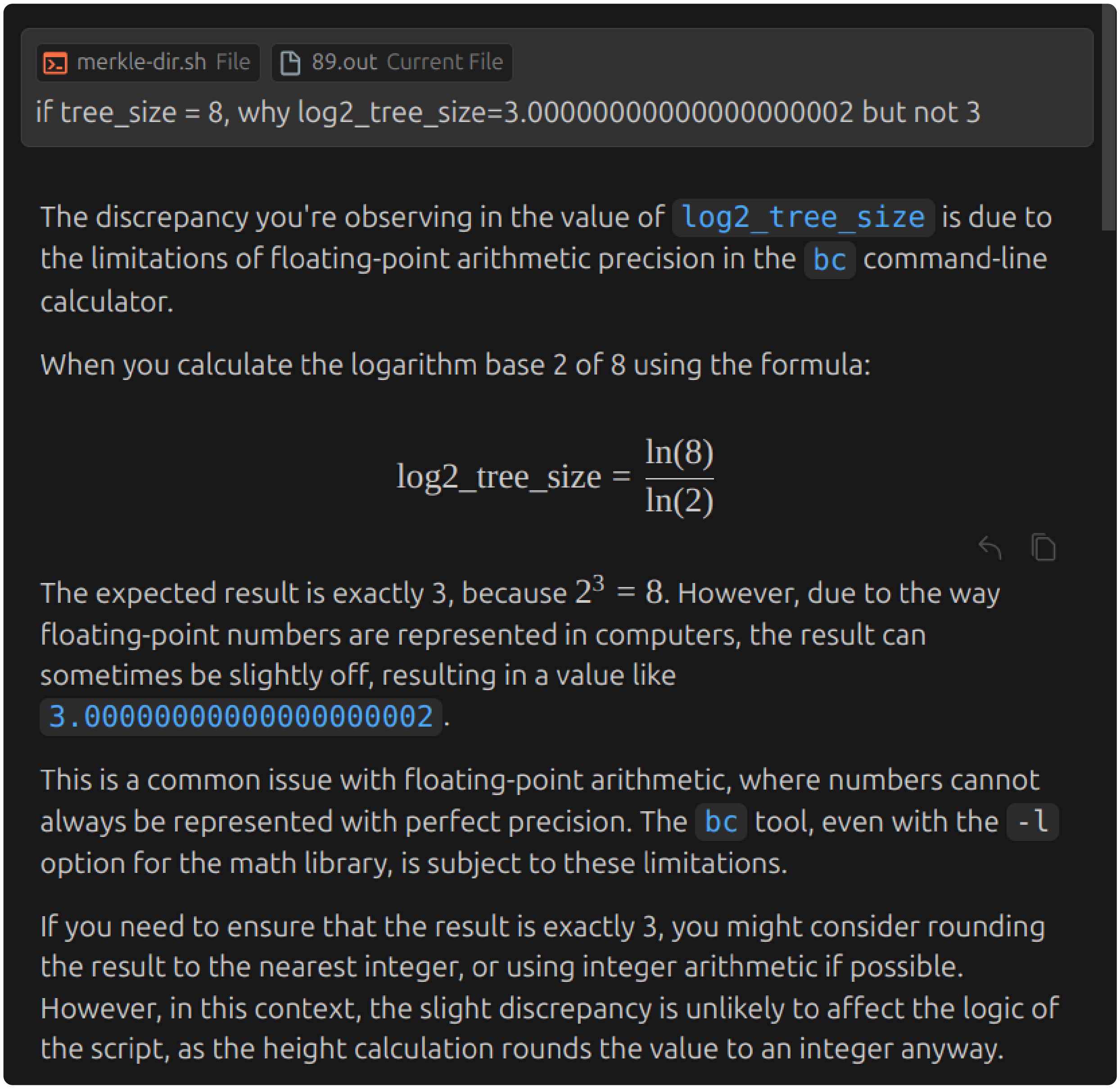
參考的網站的 URL

- https://www.baeldung.com/cs/merkle-trees
- https://www.bogotobogo.com/Linux/linux_File_Types.php
- https://docs.google.com/presentation/d/1pcLXptV4PBxsH-T35E0CCAu5jakcaXj02s4vphmS7p8/edit#slide=id.g325b2d5acdd_0_13
- https://ithelp.ithome.com.tw/articles/10327486

與我討論的同學

- B11902194 賴睿廷

我與生成式 AI 的完整對話截圖



if tree_size = 8, why log2_tree_size=3.00000000000000000002 but not 3

The discrepancy you're observing in the value of `log2_tree_size` is due to the limitations of floating-point arithmetic precision in the `bc` command-line calculator.

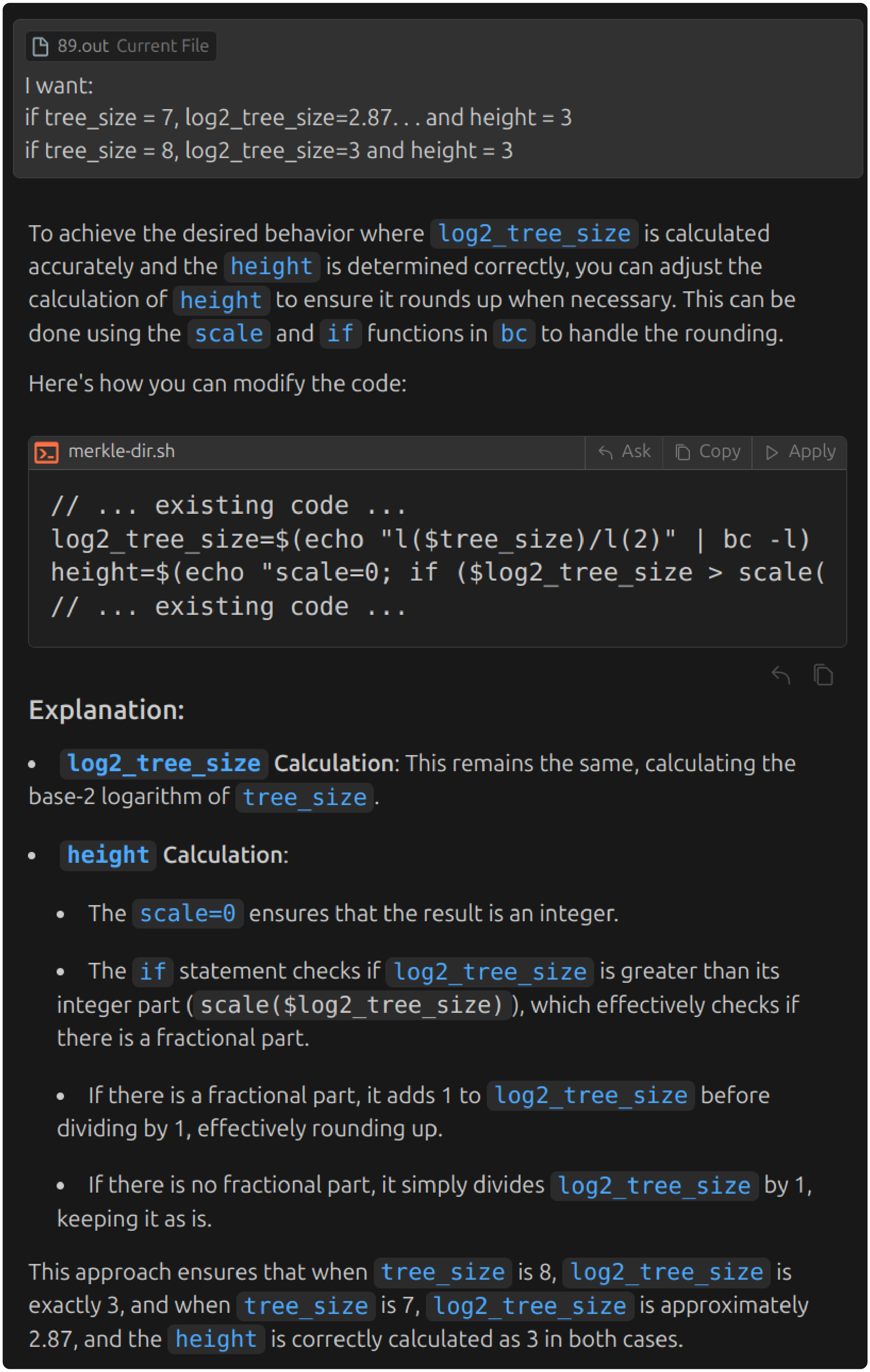
When you calculate the logarithm base 2 of 8 using the formula:

$$\text{log2\_tree\_size} = \frac{\ln(8)}{\ln(2)}$$

The expected result is exactly 3, because $2^3 = 8$. However, due to the way floating-point numbers are represented in computers, the result can sometimes be slightly off, resulting in a value like `3.00000000000000000002`.

This is a common issue with floating-point arithmetic, where numbers cannot always be represented with perfect precision. The `bc` tool, even with the `-l` option for the math library, is subject to these limitations.

If you need to ensure that the result is exactly 3, you might consider rounding the result to the nearest integer, or using integer arithmetic if possible. However, in this context, the slight discrepancy is unlikely to affect the logic of the script, as the height calculation rounds the value to an integer anyway.



I want:
if tree_size = 7, log2_tree_size=2.87. . . and height = 3
if tree_size = 8, log2_tree_size=3 and height = 3

To achieve the desired behavior where `log2_tree_size` is calculated accurately and the `height` is determined correctly, you can adjust the calculation of `height` to ensure it rounds up when necessary. This can be done using the `scale` and `if` functions in `bc` to handle the rounding.

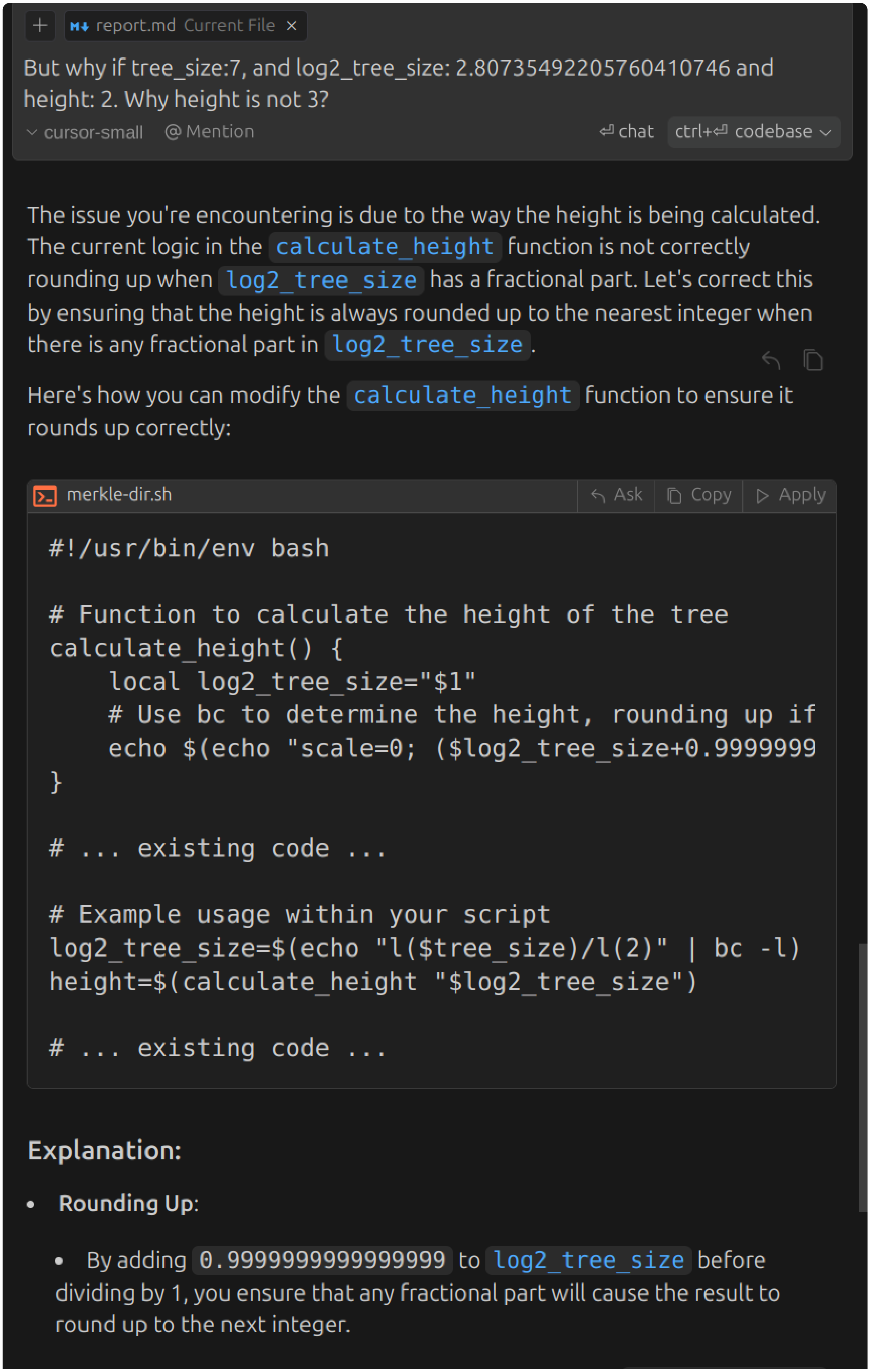
Here's how you can modify the code:

```
// ... existing code ...
log2_tree_size=$(echo "l($tree_size)/l(2)" | bc -l)
height=$(echo "scale=0; if ($log2_tree_size > scale(
// ... existing code ...
```

Explanation:

- **`log2_tree_size` Calculation**: This remains the same, calculating the base-2 logarithm of `tree_size`.
- **`height` Calculation**:
  - The `scale=0` ensures that the result is an integer.
  - The `if` statement checks if `log2_tree_size` is greater than its integer part (`scale($log2_tree_size)`), which effectively checks if there is a fractional part.
  - If there is a fractional part, it adds 1 to `log2_tree_size` before dividing by 1, effectively rounding up.
  - If there is no fractional part, it simply divides `log2_tree_size` by 1, keeping it as is.

This approach ensures that when `tree_size` is 8, `log2_tree_size` is exactly 3, and when `tree_size` is 7, `log2_tree_size` is approximately 2.87, and the `height` is correctly calculated as 3 in both cases.



But why if tree_size:7, and log2_tree_size: 2.80735492205760410746 and height: 2. Why height is not 3?

The issue you're encountering is due to the way the height is being calculated. The current logic in the `calculate_height` function is not correctly rounding up when `log2_tree_size` has a fractional part. Let's correct this by ensuring that the height is always rounded up to the nearest integer when there is any fractional part in `log2_tree_size`.

Here's how you can modify the `calculate_height` function to ensure it rounds up correctly:

```
#!/usr/bin/env bash

# Function to calculate the height of the tree
calculate_height() {
    local log2_tree_size="$1"
    # Use bc to determine the height, rounding up if
    echo $(echo "scale=0; ($log2_tree_size+0.9999999
}

# ... existing code ...

# Example usage within your script
log2_tree_size=$(echo "l($tree_size)/l(2)" | bc -l)
height=$(calculate_height "$log2_tree_size")

# ... existing code ...
```

Explanation:

- **Rounding Up**:
  - By adding `0.9999999999999999` to `log2_tree_size` before dividing by 1, you ensure that any fractional part will cause the result to round up to the next integer.

- This approach effectively rounds up any non-integer `log2_tree_size` to the nearest whole number, ensuring that the height is calculated correctly.

With this change, when `tree_size` is 7, `log2_tree_size` will be approximately 2.807, and the `height` will be correctly calculated as 3.