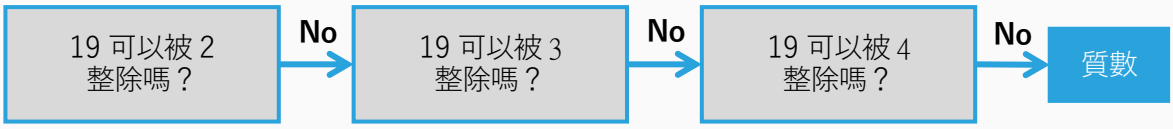
3.1 節末問題 3.1 的解答

問題 3.1.1

對於自己的年齡 N 為 2 歲以上的情況，可以進行如下判斷：

- 若無法被 2 以上 $\sqrt{N}$ 以下的整數整除時：質數
- 若無法被整除：合數

例如 19 歲的情況，根據 $\sqrt{19} = 4.358\ldots$，若可以被 $2, 3, 4$ 整除即為質數，但因為都無法被整除，所以19為質數。



問題 3.1.2

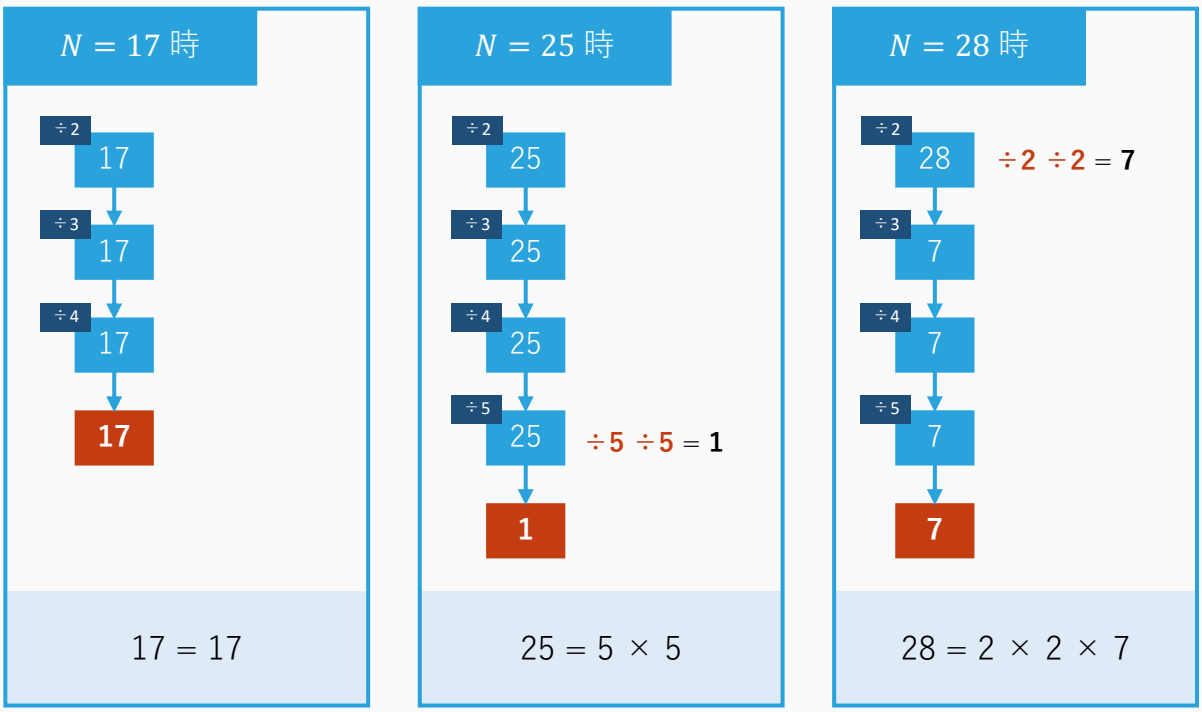
首先，當將自然數 N 進行質因數分解時，超過 $\sqrt{N}$ 的質因數最多只有一個。這可以用反證法（→**3.1.3項**）如下證明。

> 假設將自然數 N 進行質因數分解時，超過$\sqrt{N}$ 的質因數有 2 個以上。即，假設用超過 N 的整數 A 和 B 來進行質因數分解，如下所示：。
>
> $$N = ○\times○\times\cdots\times○\times A\times B$$
>
> 然而，因為 $A\times B > N$ 而產生矛盾。因此，超過 $\sqrt{N}$ 的質因數最多只有一個。

因此，可以用以下的演算法來進行質因數分解：

- 將 N 盡可能多次地除以 2 。
- 將 N 盡可能多次地除以 3 。
- 對 $4, 5, \ldots, \lfloor\sqrt{N}\rfloor$ 進行同樣的操作。。
- 最後，若剩下的 N 不為 1 時，則將其加入質因數。

例如，於 $N = 17, 25, 28$ 的情況，演算法將如下圖所示運作。注意可能會有同一數字被多次除的情況。



C++ 的實作例如下。

```cpp
#include <iostream>
using namespace std;

int main() {
    // 輸入
    long long N;
    cin >> N;

    // 質因數分解、輸出
    for (long long i = 2; i * i <= N; i++) {
        while (N % i == 0) {
            N /= i;
            cout << i << endl;
        }
    }
    if (N >= 2) cout << N << endl;
    return 0;
}
```

※ Python 等原始碼請參閱 chap3-1.md 。