

Operating Systems-1: Autumn 2023

Programming Assignment 1: Finding Tetrahedral Numbers

Waghmare Aditya Abhaykumar
CS22BTECH11061

November 30, 2023

1 Coding Approach

Program creates `K` child processes to check if the numbers assigned to them are tetrahedral numbers or not and communicate the result back to the parent/main process.

1.1 Parent/Main process

The Main process retrieves the parameters `N` and `K` from the input file. For this purpose, it uses the `get_input(int *n, int *k)` function. Once retrieved, we check if `N > 0` and `K > 0`.

When `N` and `K` values are valid, we create `K` child processes of the Main process, which check if the numbers assigned to them are tetrahedral and store the result in `K` shared memory buffers.

The naming convention for these shared memory buffers is `'shared_memory_buffer<child_number>'` where `child_number` ranges from 0 to `K-1`.

Shared memory buffers use `bool` type variables to store the results. Where each index corresponds to some number from 1 to `N`. The assignment of numbers is done based on the modulo operation.

For Child `i`, number `n` is assigned to it such that $(n \bmod k) = (i+1)$ for `i = 0` to `K-2` and $(n \bmod k) = 0$ for `i = K-1`, where `i` is from 0 to `K-1`.

The Main process waits for all the child processes to terminate using `wait(NULL)`. After the Main process is done waiting, it accesses the shared memory buffers and outputs the result from it in 'Out-Main' file.

1.2 Child Process

Each child process creates shared memory buffer of size `ceil(N/K)` to store its results. It checks if a number assigned to it is Tetrahedral number or not using `ifTetrahedral(int num)` function.

If a number is tetrahedral it stores `true` on that numbers respective index in shared memory buffer. It also writes the result in its output file. Once this is done the child process terminates itself.

1.2.1 ifTetrahedral(int num)

This function works like binary search to check if a number is present in the set of tetrahedral numbers or not. If 'num' is tetrahedral it returns `true` else `false`. We have a function `pth_tetrahedral(long long p)` which returns the `p`th tetrahedral number, i.e. for `p = 1` result is 1, for `p = 2` result is 4, for `p = 3` result is 10, and so and so. `pth_tetrahedral(long long p)` function can be thought of as a sorted array where index is represented by `p`.

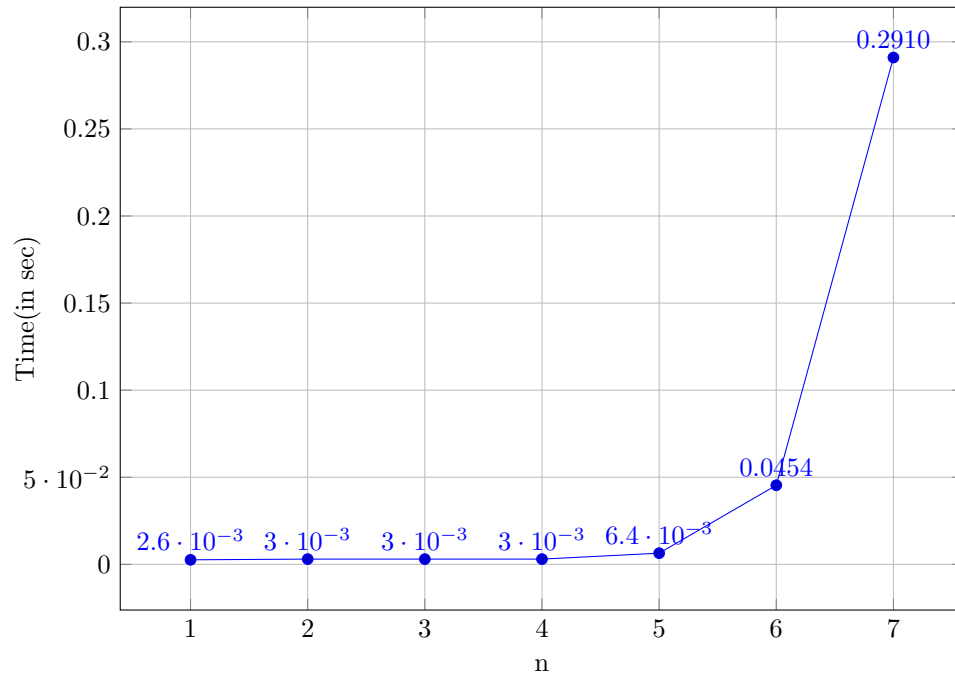
Hence, in `ifTetrahedral(int num)` we perform binary search for 'num' with `pth_tetrahedral(long long p)` function. For this purpose we pick `start = 1` and `end = p_max` such that `p_max` is the index of largest tetrahedral number less than `INT_MAX` (from `limits.h`). To compute `p_max` we have a function `set_p_max()`, which sets the value of global `p_max` accordingly. Therefore, we need to call the function `set_p_max()` before using `ifTetrahedral(int num)` function.

2 Output Time Analysis

2.1 Time vs Size, N:

$K = 8$

n is such that $N = 2^{3n}$



2.2 Time vs Number of Processes, K:

$N = 1000000$

