

## 1 19-04-18

- Coba cek program pake Python. Kalo bisa, baru ke C dkk.
- vi cprograme :visual
- array in c++ is nicer, ga perlu define alloc dkk kayak di C
- tergantung Processor juga, jadi bukan karena C aja, bisa karena besarnya processor
- The -O3 optimization : dia bisa tahu variabel apa yang dipake dan ngga. Jadi kalo ada variabel yang ga kepake untuk selanjutnya, di skip. Jadi karena itu dia lebih cepat. kalo 1jt di jumlah terus di print, 1/3 Processor harusnya waktunya.
- 1 addition take 3 cycle
- komp cuma simpen finite number, sekitar 50/15 digit. Jadi kita bisa kehilangan info, misalnya dengan asosiatif.
- komputer ngubah float ke binary, bisa jadi 0.1 ga sama. ngubah urutan operasi bisa jadi jawaban yang keluar ga sama
- kalo pake kurung, bisa jadi lebih efektif. karena 4 hal yang dijumlahkan bisa 2 kali dikerjakan dalam satu waktu
- -ffast-math bisa save 2/3 dr -O3, karena re order opertaion itself, tapi kudu ati2. Jadi kalo -O3 dan -O3 -ffast-math sama, berarti program uda max. ada lagi -march=native bikin lebih cepat, karena works di komp kita sendiri , in new computer is helping karena ada banyak command baru ?.
- coba profiling, jadi biar kelihatan bagian mana dari code yang lama.
- running cuma di 1 core. Kalo mau parallel code, lebih complicated, pake IMD ?