Problem set 1

Peter Rotar

September 2023

Question 1: The von Neumann bottleneck happens because of the speed disparity between CPU and RAM, i.e. CPU has to "wait" for the data from RAM. To resolve this situation, one of the solutions is to implement caches, which reduce CPU waiting time.

Question 2: In the following code

```
int a[100], b[100];
for (int i = 0; i < 100; i++) {
    b[i] = i;
    a[i] = 2*b[i];
    b[i] = i*i;
}
```

we see that there is data dependency since in the fourth row list **a** is dependent of the data from the list **b**. It is also control dependent since it is dependent on the condition $i < 100$ from the *for* loop, i.e the *for* loop needs to finish first before continuing. It is not name dependent, since there are no "clashes" between third and fifth row, i.e. the value from the fifth row the list **b** gets, has no purpose in the following iterations.

Question 3: In SIMD each instruction executes on multiple data in some order, i.e. there is a dependent sequence of instructions, where the ith instruction executes when the $(i-1)$th instruction finishes its execution. In MIMD each instruction works independently, meaning we can have multiple instructions that can execute on the same multiple data at the same time.

Question 4: For documenting computational work and execution time, I took variables from the example. The following results are:

Solvers	computational work (iterations)	elapsed time (s)
Jacobi	3869716	13.102394
Gauss-Seidel	3079497	52.507077
R-B Gauss-Seidel	2826014	9.957485

As seen from the table, Gauss-Seidel method has less computational work, but more elapsed time. This is because the method is an upgrade of Jacobian, which computes the i-th component of k-th iteration with already computed components from the same k-th iteration. Hence it uses less computation, but more time since new approximations are dependent. Therefore, unlike in the Jacobian method, it can not use parallelism. This problem is mitigated with modified Gauss-Seidel, so called Red-Black Gauss-Seidel which also uses parallelism.

Question 5:

```
model name      : AMD Ryzen 5 5600H with Radeon Graphics
```