Mindy Mohr
917425752
ECS36B HW4 Part 1

Running the UNIX time command of the copyij and copyji functions on my local machine, I found copyij to be overall faster in user, system, CPU, and total times than copyji:

```
    Mindy@matts-air HW4 % time ./hw4part1_ij.o
    ./hw4part1_ij.o  0.02s user 0.02s system 85% cpu 0.053 total
    Mindy@matts-air HW4 % time ./hw4part1_ji.o
    ./hw4part1_ji.o  0.15s user 0.03s system 96% cpu 0.186 total
```

I also ran the functions again on the CSIF machines and confirmed that copyij is still overall faster than copyji, except this time with real time instead of CPU and total times:

```
    memohr@ad3.ucdavis.edu@pc35:~$ time ./hw4part1_ij.o
    real 0m0.022s
    user 0m0.012s
    sys 0m0.008s
    memohr@ad3.ucdavis.edu@pc35:~$ time ./hw4part1_ji.o
    real 0m0.081s
    user 0m0.059s
    sys 0m0.020s
```

The difference in times is due to the order of the nested loops and the elements of the 2D arrays. Function copyij runs faster because the order of elements in an array impacts the memory by row-major order, and the machine does not have to access the main memory as often. Since “i” is the row element in the 2D arrays and the variable condition for the outer loop in copyij, the machine will conveniently access the adjacent memories such as “i+1” along with the “i” memory, making the process faster.

On the other hand, because “j” is the variable condition in copyji when “i” is the row element, the machine has to access the main memory more often, making the process slower. As a result, when I switched the order of the elements in the 2D arrays, so making “j” the row element, dst[j][i] = src[j][i] rather than dst[i][j] = src[i][j], I found copyji to be faster in this case:

```
    Mindy@matts-air HW4 % time ./hw4part1copy_ij.o
    ./hw4part1copy_ij.o  0.16s user 0.04s system 42% cpu 0.457 total
    Mindy@matts-air HW4 % time ./hw4part1copy_ji.o
    ./hw4part1copy_ji.o  0.15s user 0.03s system 95% cpu 0.195 total

    memohr@ad3.ucdavis.edu@pc35:~$ time ./hw4part1copy_ij.o
    real 0m0.080s
    user 0m0.061s
    sys  0m0.017s
    memohr@ad3.ucdavis.edu@pc35:~$ time ./hw4part1copy_ji.o
    real 0m0.030s
    user 0m0.016s
    sys  0m0.012s
```

In conclusion, in the original program, the copyij function is faster than the copyji function for both machines since “i” is the row element in the 2D arrays and the variable condition in the outer loop of the function.