

See [homework 1](#) for more instructions on handing in homework.

1. **Pthreads implementation of dynamic programming (50 points)**. Implement a function:

```
int64 pthreadedist(String *top, String *left, Penalty *px, int nThreads)
```

that computes the editing distance between `top` and `left` using the insert, delete, and replace costs as specified by `px`, and uses `nThreads` (POSIX pthreads) threads. The return value is the editing distance between the strings.

- Your implementation should be based on the `mpiedist` function from `mpiedist.c`. Modify the code to use pthreads instead of MPI.
- I will release a revised version of `time_edist` that will call `pthreadedist`. Your code must work with this implementation of `time_edist`.
- I will release a `Makefile` to be used to build your program. You can modify it for code development. To receive full credit, your code must build with no errors or warnings using the version that I provide.

Finally, grading. I will write a robot to grade your solution on a scale of 0 to 50. I'll probably release the code for the robot so you can use it when developing your code. I will either replace your lowest homework score (by percentage) with the percentage score that you get on this assignment, or I will give you the raw score of this assignment as bonus homework marks – whichever one is better for your grade.