

IN3200/IN4200 Exercise Set 9

Exercise 1: Work division

(a) A concrete case

There are in total 20 identical tasks which are independent of each other. There are in total 6 workers where three of them are “regular workers” and the other three are “super workers”. A regular worker needs two days to finish a task, whereas a super worker only needs one day. (We assume that a task can only be executed by one worker.)

How many days in minimum do the six workers need to finish the 20 tasks?

(b) Generalization

Assume there are m identical tasks which are independent of each other. There are p super workers and p regular workers. As before, a regular worker needs two days to finish a task, whereas a super worker only needs one day. (We also assume that a task can only be executed by one worker.)

How many days in minimum need these $2p$ workers to finish the m tasks?

Exercise 2: Find error(s) and improve

There is one (or several) error(s) in the following OpenMP code. Please correct the error(s). Is there anything that can be further improved?

```
int i, N=100000;
double u[N], v[N];

for (i=0; i<N; i++) {
    u[i] = 0.001*(i-N/2);
    v[i] = 0.0;
```

```

}

#pragma omp parallel default(shared)
{
    int time_step;
    double *tmp;

    for (time_step=0; time_step<1000; time_step++)
    {
        #pragma omp for nowait
        for (i=1; i<N-1; i++)
            v[i] = u[i-1]-2*u[i]+u[i+1];

        tmp = v;
        v=u;
        u=tmp;
    }
}

```

Exercise 3: OpenMP parallelization

We have a serial code as below. Please parallelize the code with OpenMP.

```

int i, N = 100000;
double s = 0.;

double *a = (double*)malloc (N*sizeof(double));
double *b = (double*)malloc (N*sizeof(double));

for (i=0; i<N; i++) {
    a[i] = 10.0+2*i;
    b[i] = 20.0+sin(0.1*i);
}

for (i=0; i<N-1; i++) {
    s += a[i];
    a[i+1] = cos(b[i]);
}

```

Exercise 4: Read a web graph stored in file

Suppose a so-called web graph is stored in a text file with the follow format:

- The first two lines both start with the # symbol and contain free text (listing the name of the data file, authors etc.);
- Line 3 is of the form “# Nodes: integer1 Edges: integer2”, where integer1 is the total number of webpages, and integer2 is the total

number of links. (Here, nodes mean the same as webpages, and edges mean the same as links.)

- Line 4 is of the form “# FromNodeId ToNodeId”;
- The remaining part of the file consists of a number of lines, the total number equals the number of links. Each line simply contains two integers: the index of the outbound webpage and the index of the inbound webpage;
- Some of the links can be self-links (same outbound as inbound), these should be excluded when creating the hyperlink matrix;
- Note: the webpage indices start from 0 (C convention).

Please write a C program that can accept on the command line the filename of a web graph and find out the following info:

- The webpage(s) with the most inbound links;
- The webpage(s) with the most outbound links;
- The number of self-links (same outbound as inbound), if any.

As an example, please test your implementation on web graph file that can be downloaded from

<https://snap.stanford.edu/data/web-NotreDame.html>