Parallel Programming

CS575

Chao Zhang

Project #4

1. Source listing

project4.cpp

```
const float AVG_TEMP                = 50.0;
const float AMP_TEMP                = 20.0;
const float RANDOM_TEMP             = 10.0;

const float MIDTEMP                 = 40.0;
const float MIDPRECIP               = 10.0;
const float MIDLIGHTDUR             = 100.;

const float RANDOM_OVERCAST         = 8.0;

void DeerHunter() {
    while(NowYear <= 2021) {
        int tempNumHunter = NowNumHunter;
        int tempNumDeer = NowNumDeer;
        float huntFactor = 1.;
        if (tempNumDeer == tempNumHunter)
            huntFactor = (float)tempNumDeer * (float)tempNumHunter / (float)tempNumHunter;
        else
            huntFactor = (float)tempNumHunter * (float)tempNumDeer / (float)abs(tempNumDeer - tempNumHunter);
        int huntNum = (int)(Ranf(0., huntFactor));


        if (tempNumHunter == 0 && tempNumDeer >= 3)
            tempNumHunter = 1;
        else if (tempNumHunter > 0 && tempNumDeer >= 3) {
            if (huntNum >= 1) {
                tempNumHunter ++;
                tempNumDeer -= huntNum;
            }
            else {
                tempNumHunter --;
            }
        }
        else if (tempNumHunter != 0 && tempNumDeer < 3)
            tempNumHunter --;
        #pragma omp barrier

        if (tempNumHunter < 0)
            tempNumHunter = 0;
        NowNumHunter = tempNumHunter;
```

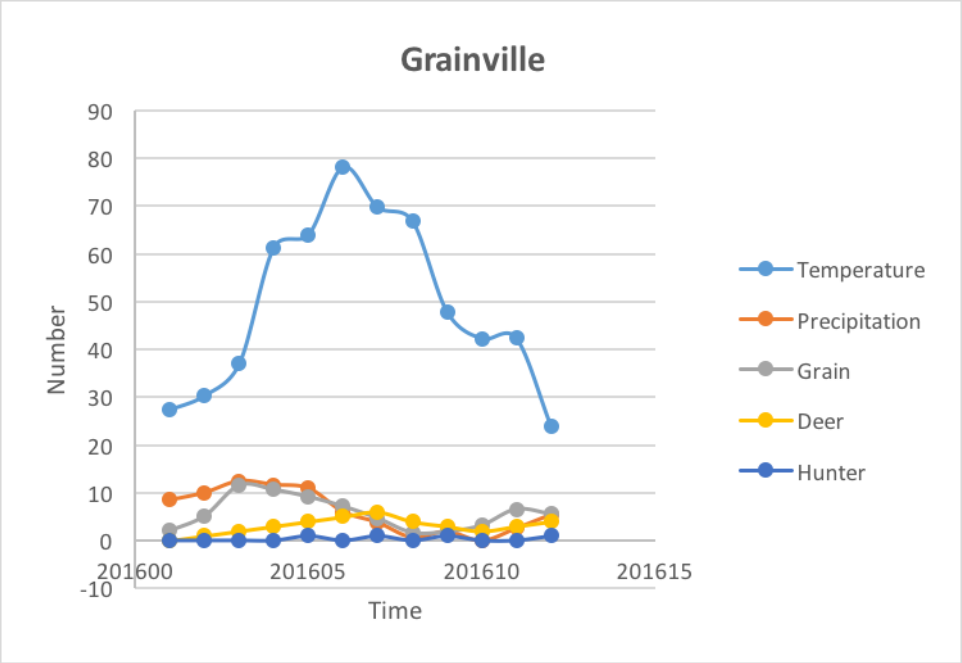
This is the code of myagent, I named my agent as the DeerHunter. The hunter will start hunting the deer when there has three or more than three deer. The hunter has the same number as the deer in the beginning. Like the real, the number of hunters will decreased by one if they didn't hunt any deer in a month because of the 0 income.

2. Result and analysis

Because of the data is to long in one table, so I make them separated by year.

Data	Temperature	Precipitation	Grain	Deer	Hunter
201601	27.293438	8.466307	2.054808	0	0
201602	30.267059	9.99987	5.157097	1	0
201603	36.964581	12.405725	11.542672	2	0
201604	61.351582	11.634737	10.624251	3	0
201605	63.983932	10.892292	9.149454	4	1
201606	78.207031	6.008617	7.149457	5	0
201607	69.771072	3.778853	4.650226	6	1
201608	66.877823	0.666838	1.652666	4	0
201609	47.895382	1.927604	1.888099	3	1
201610	42.225563	0	3.188914	2	0
201611	42.32209	2.577969	6.558442	3	0
201612	23.980442	5.406227	5.556097	4	1

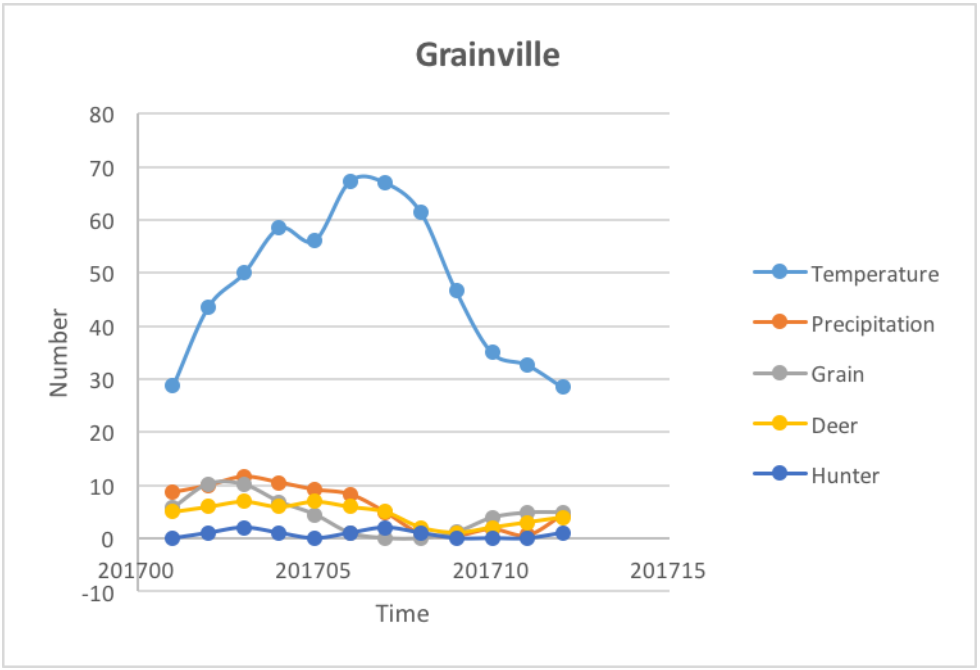
Table for 2016



Graph for 2016

Data	Temperature	Precipitation	Grain	Deer	Hunter
201701	28.725445	8.653484	5.759834	5	0
201702	43.639229	9.94244	10.267246	6	1
201703	49.981457	11.555356	10.150604	7	2
201704	58.522774	10.479457	6.908864	6	1
201705	56.177444	9.166726	4.488938	7	0
201706	67.352325	8.273807	0.993313	6	1
201707	67.026031	4.947616	0	5	2
201708	61.440491	0.939847	0	2	1
201709	46.591667	0.561126	1.125522	1	0
201710	35.106022	1.821658	3.850991	2	0
201711	32.742935	0.64508	4.820229	3	0
201712	28.539598	4.407367	4.893659	4	1

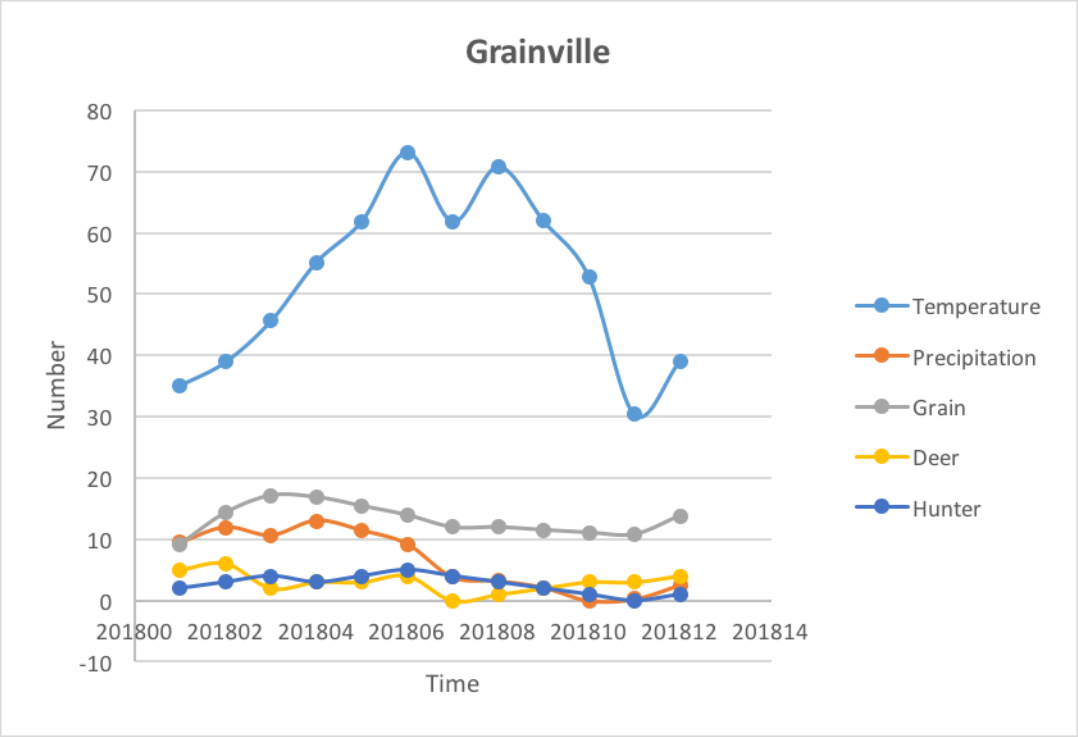
Table for 2017



Graph for 2017

Data	Temperature	Precipitation	Grain	Deer	Hunter
201801	35.103313	9.451004	9.169152	5	2
201802	38.952774	11.904806	14.299946	6	3
201803	45.637104	10.613775	17.100218	2	4
201804	55.179916	12.986778	16.830688	3	3
201805	61.881653	11.388431	15.396044	3	4
201806	73.133537	9.162123	13.896179	4	5
201807	61.803169	3.80627	11.943163	0	4
201808	70.739456	3.203041	11.94356	1	3
201809	61.895233	2.117817	11.479144	2	2
201810	52.701309	0	11.065522	3	1
201811	30.377403	0.242373	10.788618	3	0
201812	39.062668	2.498522	13.806005	4	1

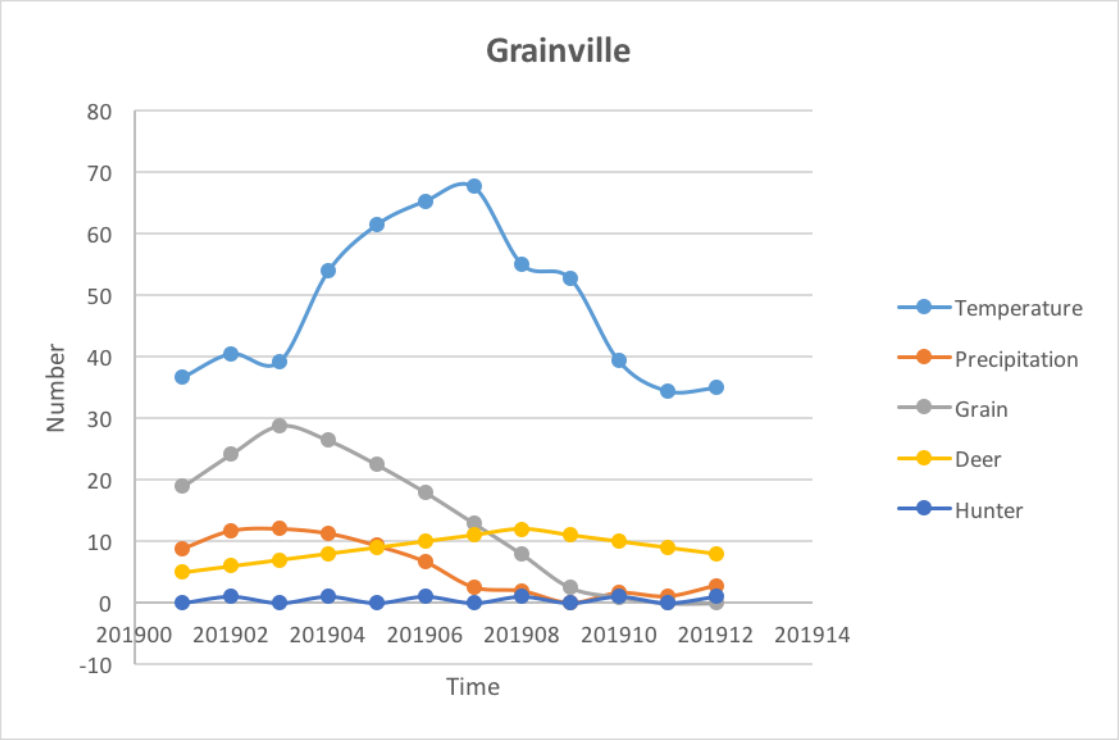
Table for 2018



Graph for 2018

Data	Temperature	Precipitation	Grain	Deer	Hunter
201901	36.66761	8.864387	18.873438	5	0
201902	40.461269	11.763597	24.111963	6	1
201903	39.224327	12.082109	28.726606	7	0
201904	53.837555	11.328956	26.384962	8	1
201905	61.376003	9.341034	22.467522	9	0
201906	65.238922	6.714912	17.979816	10	1
201907	67.553581	2.568235	12.98214	11	0
201908	55.005058	1.93834	7.9217	12	1
201909	52.638645	0	2.517464	11	0
201910	39.307976	1.634728	0.972045	10	1
201911	34.442055	1.074266	0	9	0
201912	34.933025	2.827144	0	8	1

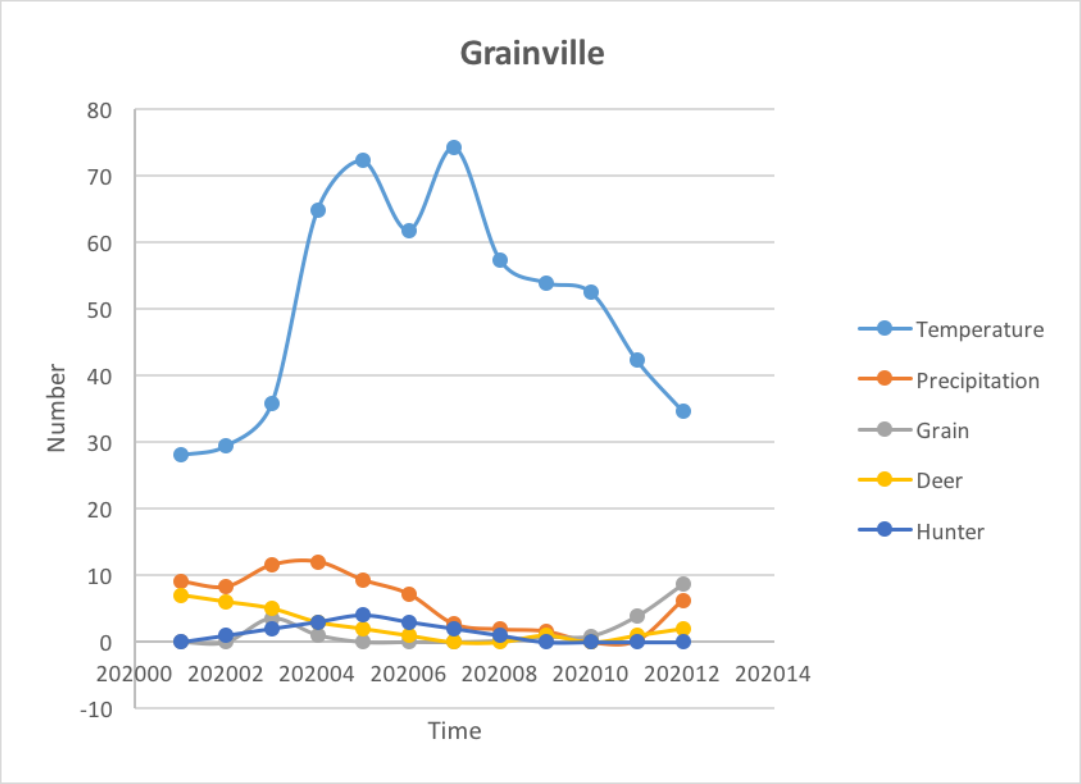
Table for 2019



Graph for 2019

Data	Temperature	Precipitation	Grain	Deer	Hunter
202001	28.073738	9.116754	0	7	0
202002	29.487047	8.32258	0	6	1
202003	35.829071	11.538677	3.56531	5	2
202004	64.79689	12.039042	1.081702	3	3
202005	72.239151	9.316163	0	2	4
202006	61.837547	7.213348	0	1	3
202007	74.205223	2.716992	0	0	2
202008	57.391956	1.952154	0.203307	0	1
202009	53.969547	1.648569	0.769112	1	0
202010	52.443939	0	0.894693	0	0
202011	42.280815	0.310775	3.864834	1	0
202012	34.666	6.289401	8.609653	2	0

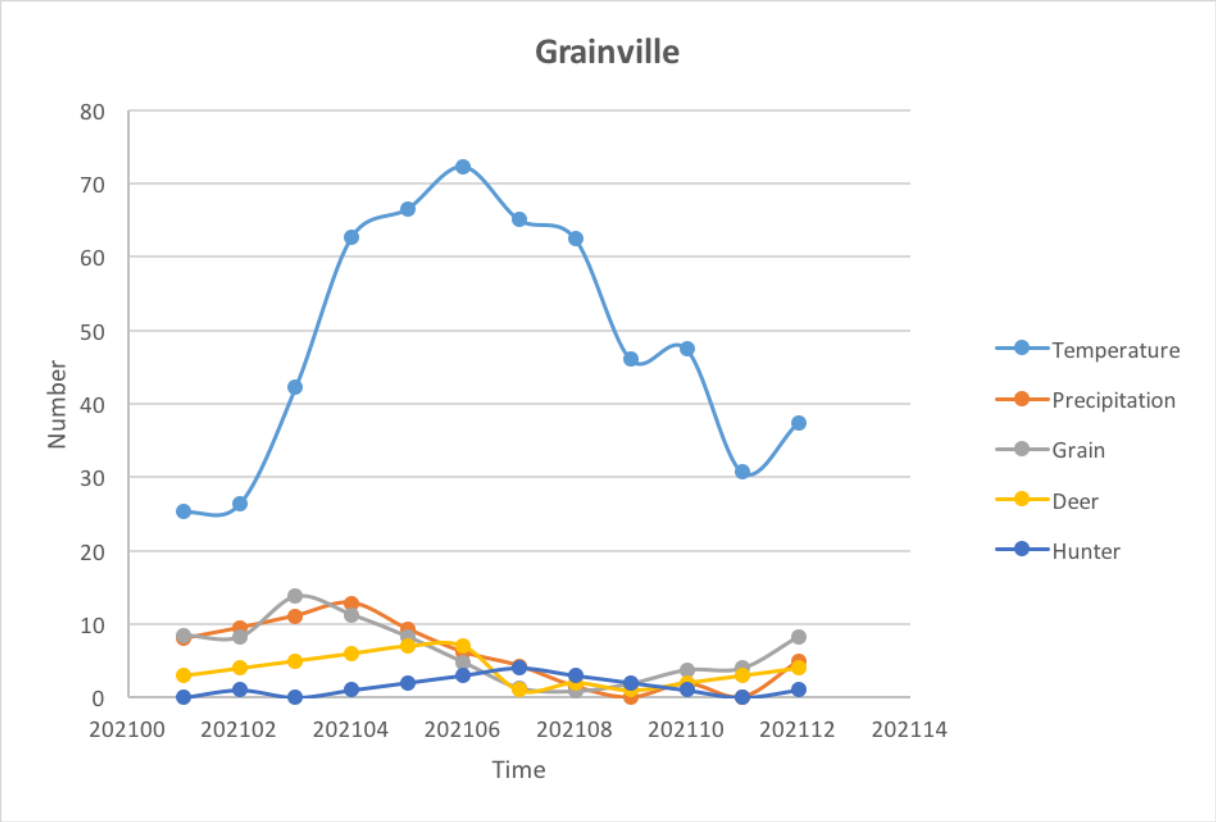
Table for 2020



Graph for 2020

Data	Temperature	Precipitation	Grain	Deer	Hunter
202101	25.260666	8.007852	8.485358	3	0
202102	26.434719	9.468509	8.2521	4	1
202103	42.302494	11.105555	13.746861	5	0
202104	62.700348	12.840648	11.289528	6	1
202105	66.588219	9.30774	8.296302	7	2
202106	72.351349	6.158131	4.796498	7	3
202107	65.111679	4.28435	1.307032	1	4
202108	62.540031	1.482454	0.831107	2	3
202109	46.076538	0	1.865501	1	2
202110	47.475071	1.859787	3.724044	2	1
202111	30.684574	0.065835	3.976109	3	0
202112	37.449329	4.893044	8.251307	4	1

Table for 2021



Graph for 2021

For the results we can get that the height of the grain is related to the temperature, the precipitation, and the number of deer. It goes down with the growth of the number of deer, almost the same as the precipitation, and can be highest with a specified temperature. The number of hunter is related to the number of deer. When we have many deer, the hunter will increase and with the increase of hunter, the number of deer goes down. All of those works like this because the 3 threads work. All the thread read and write the current and the global variables as they should. So all of those factors effect each other.