MINI PROJECT COMPLEXITY RAPPORT

BELAIFA AMIR 191931029224

DJENKAL MOHAMED RACIM 181831046982

Tri à bulles

le programme C de la procédure TriBulle

```
void tribulles(int T[], int n)
{
  int x;
  int change = 1;
  while (change == 1)
  {
    change = 0;
    for (int j = 0; j < n - 1; j++)
    {
      if (T[j] > T[j + 1])
      {
        x = T[j];
        T[j] = T[j + 1];
        T[j + 1] = x;
        change = 1;
      }
    }
  }
}
```

le programme C de TriBulleOpt

```
void tribullesopt(int T[], int n)
{
  int x;
  int change = 1;
  int m=n-1;
  while (change==1)
  {
    change = 0;
    for (int j = 0; j < m; j++)
    {
      if (T[j] > T[j + 1])
      {
        x = T[j];
        T[j] = T[j + 1];
        T[j + 1] = x;
        change = 1;
      }
    }
    m=m-1;
  }
}
```

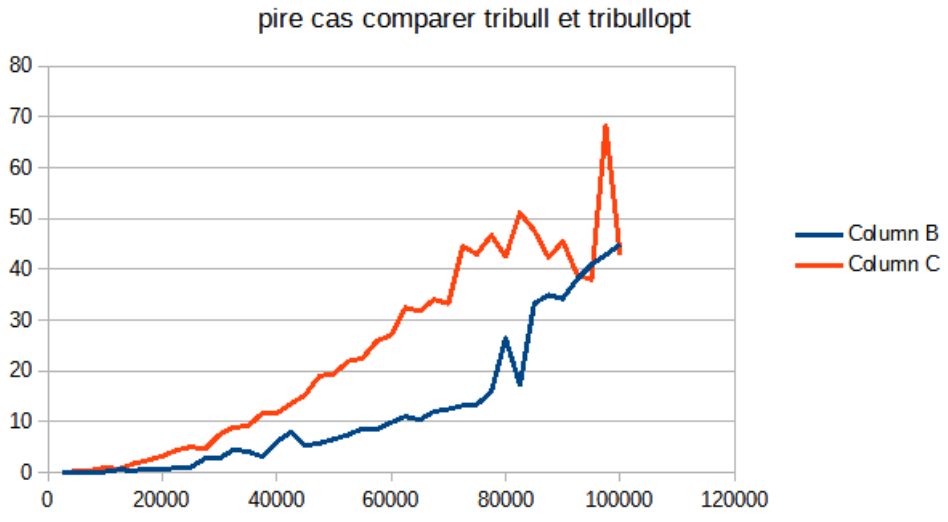
comparer Tribulle et TribulleOpt en meilleur cas

2500	1.749	1.351
5000	4.648	2.561
7500	5.418	3.743
10000	6.602	5.534
12500	10.497	6.328
15000	10.714	6.212
17500	12.657	10.496
20000	13.026	10.863
22500	20.306999	11.312
25000	20.985001	11.921
27500	26.792	13.333
30000	25.114	14
32500	28.309	15.487
35000	32.438	20.780001
37500	32.576	22.551001
40000	36.352001	24.167999
42500	33.799	25.684
45000	35.591	27.413
47500	33.976002	25.169001
50000	42.111	27.313999
52500	34.532001	28.434
55000	37.333	29.179001
57500	49.215	30.740999
60000	51.846001	32.337002
62500	46.355999	33.443001
65000	50.233002	34.793999
67500	49.671001	36.158001
70000	51.103001	37.241001
72500	53.058998	38.516998
75000	55.244999	39.729
77500	56.779999	41.577999
80000	58.522999	42.108002
82500	63.002998	43.983002
85000	66.989998	46.007
87500	58.094002	46.928001
90000	60.314999	48.308998
92500	61.678001	52.318001
95000	81.056	33.637001
97500	86.375999	300.398987
100000	81.325996	79.977997

le graph bleu est le tribull , le graph rouge est le tribullopt

comparer Tribulle et TribulleOpt en pire cas

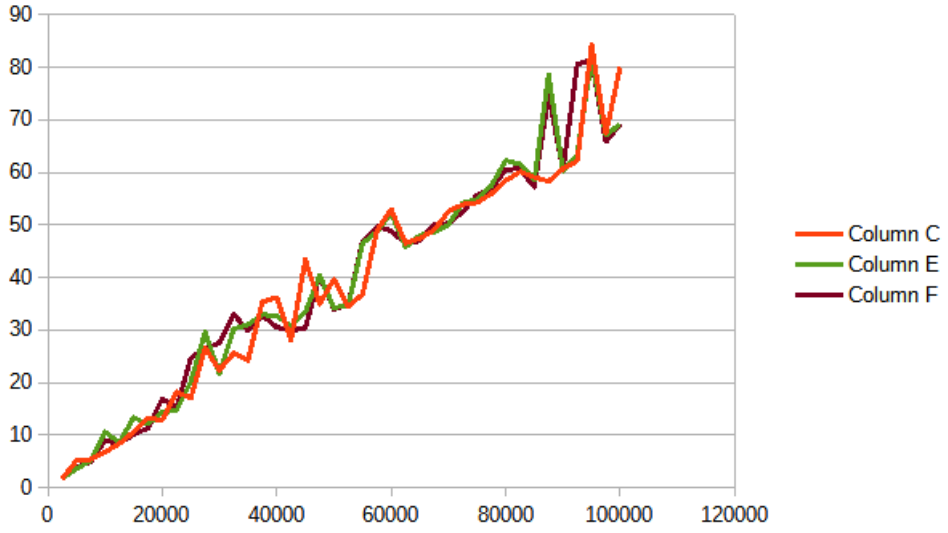
	tribullopt	tribull
2500	0.01	0.052
5000	0.043	0.218
7500	0.096	0.426
10000	0.17	0.968
12500	0.577	0.701
15000	0.383	1.761
17500	0.779	2.414
20000	0.684	3.211
22500	0.872	4.39
25000	1.069	5.057
27500	2.746	4.606
30000	2.808	7.435
32500	4.515	8.951
35000	4.115	9.098
37500	3.135	11.693
40000	6.034	11.669
42500	7.936	13.476
45000	5.303	15.32
47500	5.79	18.99
50000	6.569	19.42
52500	7.445	21.908001
55000	8.604	22.486
57500	8.546	25.882
60000	9.861	27.056
62500	11.035	32.405998
65000	10.363	31.754999
67500	12.016	34.053001
70000	12.461	33.280998
72500	13.167	44.529999
75000	13.39	42.972
77500	15.934	46.702999
80000	26.381001	42.528
82500	17.261	51.096001
85000	33.254002	47.646999
87500	34.935001	42.326
90000	34.231998	45.485001
92500	38.041	38.951
95000	40.908001	37.924999
97500	42.780998	68.202003
100000	44.890999	42.901001



le graph bleu est le tribull , le graph rouge est le tribullopt

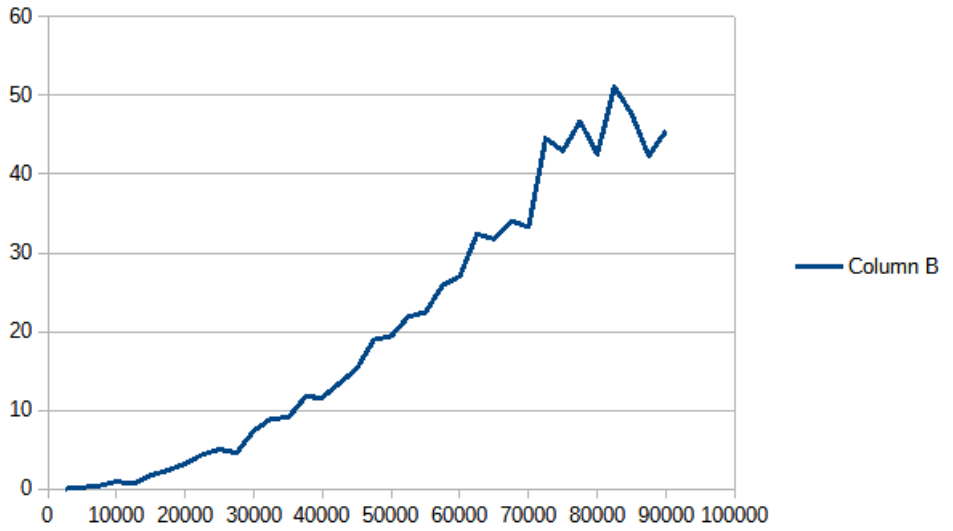
Le temp d’execution Tribulle en meilleur cas

2500	1.749	1.636	1.64	1.804	1.606
5000	4.648	5.131	3.676	3.551	3.882
7500	5.418	5.352	5.535	5.089	4.92
10000	6.602	6.736	7.582	10.587	8.985
12500	10.497	8.344	8.162	8.52	8.38
15000	10.714	10.568	14.934	13.308	10.074
17500	12.657	13.199	13.08	12.102	11.337
20000	13.026	12.829	13.687	14.35	16.917
22500	20.306999	18.202	15.132	14.717	15.194
25000	20.985001	17.004999	16.983999	20.181	24.598
27500	26.792	26.552999	25.32	29.528	26.539
30000	25.114	22.382999	23.979	21.684	27.482
32500	28.309	25.565001	31.92	30.190001	32.995998
35000	32.438	24.243	31.580999	31.052999	29.851999
37500	32.576	35.342999	29.67	32.909	32.765999
40000	36.352001	36.199001	44.841	32.695	30.516001
42500	33.799	28.033001	28.132999	30.705	29.885
45000	35.591	43.368	30.638	33.380001	30.378
47500	33.976002	34.984001	32.845001	40.389	40.276001
50000	42.111	39.630001	39.389999	34.243999	33.941002
52500	34.532001	34.426998	35.938	34.640999	34.771
55000	37.333	36.779999	57.998001	46.488998	46.900002
57500	49.215	48.863998	52.889999	48.662998	49.646999
60000	51.846001	52.984001	51.880001	52.259998	48.883999
62500	46.355999	46.516998	45.363998	45.783001	46.448002
65000	50.233002	47.408001	48.380001	48.022999	46.978001
67500	49.671001	48.987999	49.178001	48.609001	50.046001
70000	51.103001	52.571999	51.039001	50.025002	50.301998
72500	53.058998	53.870998	52.576	54.203999	52.464001
75000	55.244999	54.229	55.585999	54.883999	55.674999
77500	56.779999	55.909	56.778999	57.463001	56.737
80000	58.522999	58.424	58.257999	62.259998	60.583
82500	63.002998	60.109001	59.691002	61.548	60.673
85000	66.989998	59.076	56.473	58.555	57.181
87500	58.094002	58.310001	79.787003	78.556	74.742996
90000	60.314999	60.699001	66.775002	60.303001	60.505001
92500	61.678001	62.168999	62.844002	63.386002	80.524002
95000	81.056	84.266998	83.538002	81.764	81.456001
97500	86.375999	67.361	67.362	67.039001	65.806
100000	81.325996	80.025002	70.309998	69.172997	69.212997



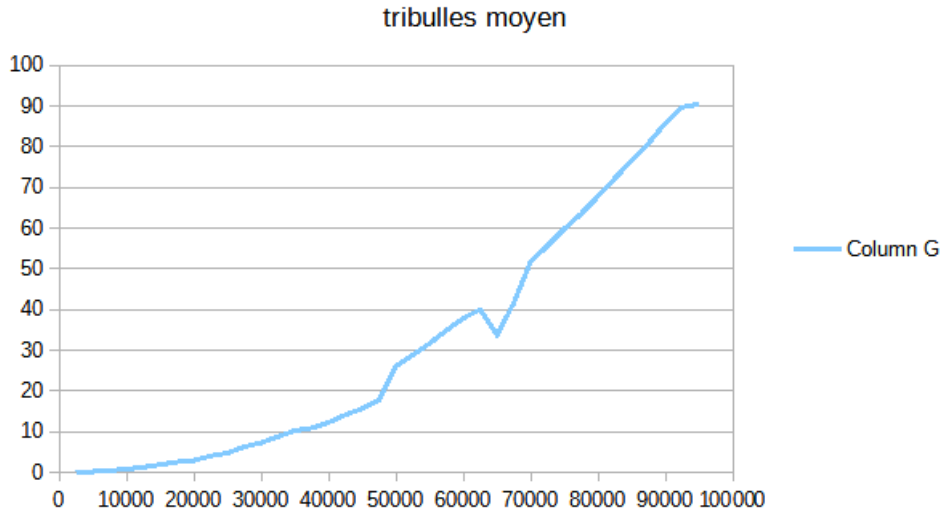
Le temp d’execution Tribulle en pire cas

2500	0.052	0.06	0.049	0.064	0.058	0.0566
5000	0.218	0.204	0.191	0.199	0.191	0.2006
7500	0.426	0.459	0.445	0.461	0.454	0.449
10000	0.968	0.848	0.794	0.347	0.344	0.6602
12500	0.701	1.216	1.229	1.226	1.32	1.1384
15000	1.761	1.856	1.881	1.987	1.743	1.8456
17500	2.414	2.452	1.874	1.797	2.467	2.2008
20000	3.211	3.214	2.202	2.768	3.107	2.9004
22500	4.39	4.366	4.225	3.011	3.741	3.9466
25000	5.057	3.571	5.509	5.098	5.253	4.8976
27500	4.606	6.306	5.816	4.516	5.747	5.3982
30000	7.435	7.609	7.591	6.253	7.748	7.3272
32500	8.951	7.138	7.256	8.35	7.023	7.7436
35000	9.098	9.152	8.408	10.197	9.213	9.2136
37500	11.693	10.287	10.805	11.734	9.609	10.8256
40000	11.669	12.486	12.932	11.767	11.699	12.1106
42500	13.476	13.193	14.506	14.355	13.013	13.7086
45000	15.32	15.467	18.177999	15.363	18.281	16.5217998
47500	18.99	19.073	17.572001	17.525999	16.627001	17.9576002
50000	19.42	19.386999	17.596001	19.464001	20.007	19.1748002
52500	21.908001	19.177	21.056999	19.813	20.556999	20.5023998
55000	22.486	21.040001	22.433001	22.065001	24.334	22.4716006
57500	25.882	26.931999	25.627001	24.399	28.030001	26.1740002
60000	27.056	28.395	29.115999	30.686001	27.476	28.5458
62500	32.405998	28.910999	32.602001	28.587	29.66	30.4331996
65000	31.754999	30.566999	31.061001	33.841999	32.245998	31.8941992
67500	34.053001	35.974998	34.391998	33.308998	33.193001	34.1843992
70000	33.280998	35.389	36.056	34.368999	35.027	34.8243994
72500	44.529999	40.164001	38.108002	37.279999	36.507999	39.318
75000	42.972	48.924	40.074001	41.849998	44.736	43.7111998
77500	46.702999	44.025002	48.891998	46.609001	44.891998	46.2241996
80000	42.528	41.771	43.354	44.219002	48.624001	44.0992006
82500	51.096001	55.042	57.167999	41.273998	49.032001	50.7223998
85000	47.646999	41.651001	42.537998	37.231998	37.958	41.4051992
87500	42.326	44.311001	43.259998	46.297001	43.576	43.954
90000	45.485001	40.699001	40.605999	40.641998	38.838001	41.254
92500	38.951	36.258999	37.223	36.009998	37.389	37.1663994
95000	37.924999	38.541	55.210999	63.133999	63.183998	51.598999
97500	68.202003	68.264999	69.335999	70.688004	59.359001	67.1700012



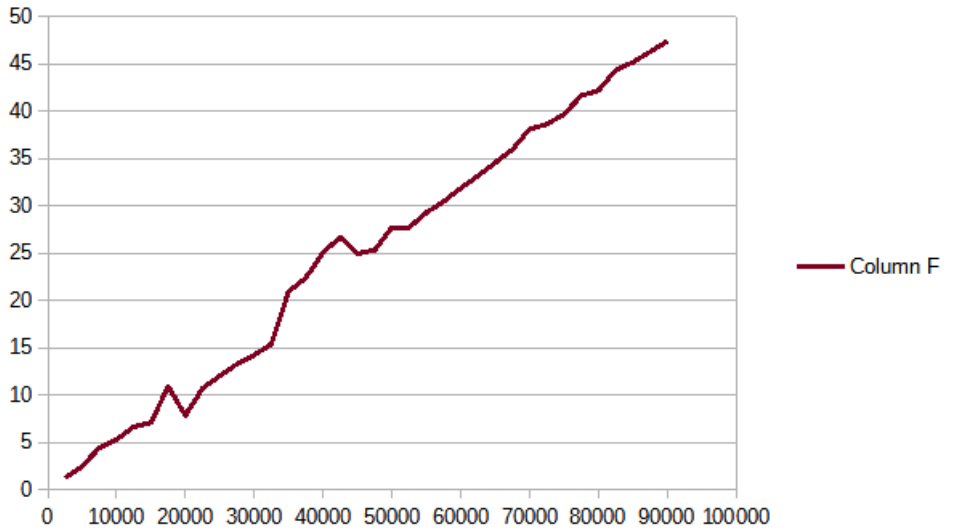
Le temp d’execution Tribulle en moyen

2500	0.042	0.046	0.043	0.039	0.044	0.0428
5000	0.176	0.186	0.173	0.19	0.177	0.1804
7500	0.433	0.426	0.47	0.43	0.432	0.4382
10000	0.794	0.807	0.794	0.846	0.785	0.8052
12500	1.221	1.278	1.287	1.253	1.304	1.2686
15000	1.964	1.921	1.8	1.91	1.877	1.8944
17500	2.527	2.603	2.489	2.577	2.579	2.555
20000	2.123	3.34	3.321	3.482	2.392	2.9316
22500	4.099	4.355	3.14	4.235	4.429	4.0516
25000	4.061	5.248	4.563	4.844	5.111	4.7654
27500	5.486	5.5	6.426	7.399	6.692	6.3006
30000	6.622	8.475	6.596	8.193	6.546	7.2864
32500	7.91	9.148	8.361	7.923	10.649	8.7982
35000	11.099	10.121	9.603	9.812	10.701	10.2672
37500	11.282	9.946	11.091	10.97	11.11	10.8798
40000	12.654	12.735	11.805	12.672	11.524	12.278
42500	14.809	13.335	14.856	13.102	14.678	14.156
45000	15.375	15.255	16.531	15.954	15.249	15.6728
47500	18.34	17.243999	18.080999	17.791	17.733	17.8377996
50000	24.923	26.358999	26.289	26.566999	26.171	26.0617996
52500	28.815001	28.910999	28.896999	28.761	28.677	28.8121998
55000	31.704	31.67	31.816999	31.76	31.545	31.6991998
57500	34.908001	34.817001	35.146999	34.778	34.687	34.8674002
60000	37.604	37.992001	37.970001	37.806999	38.054001	37.8854004
62500	41.025002	41.521999	41.000999	41.227001	35.431999	40.0414
65000	34.112	33.945999	33.972	33.584	32.786999	33.6801996
67500	35.374001	36.877998	35.543999	52.308998	48.202	41.6613992
70000	51.763	51.616001	51.919998	51.867001	51.882	51.8096
72500	55.824001	55.709	55.723999	55.667	55.526001	55.6900002
75000	59.624001	59.502998	59.801998	60.023998	60.210999	59.8327988
77500	63.227001	63.773998	63.514999	63.616001	63.678001	63.562
80000	68.031998	67.821999	67.963997	67.758003	67.568001	67.8287996
82500	72.352997	72.421997	72.322998	72.043999	72.195999	72.267598
85000	76.508003	76.726997	76.707001	76.649002	76.567001	76.6316008
87500	80.712997	81.252998	80.746002	80.584999	81.112	80.8817992
90000	85.922997	85.862	85.908997	86.130997	85.412003	85.8473988
92500	90.518997	90.818001	90.778999	90.160004	86.352997	89.7257996
95000	90.492996	91.013	90.600998	89.859001	90.242996	90.4417982



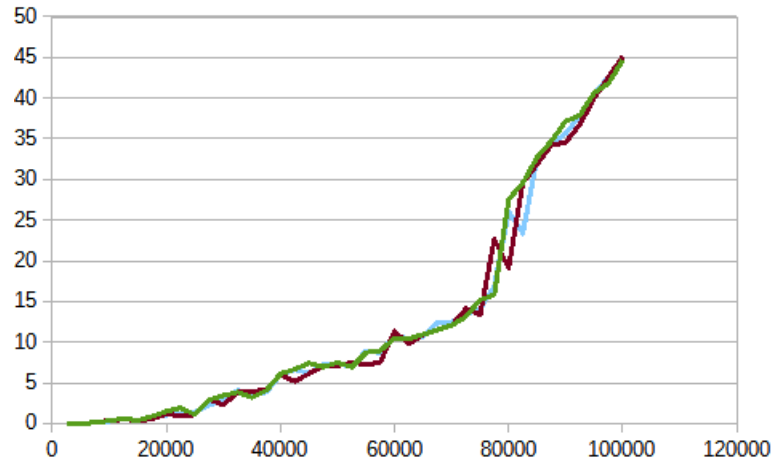
Le temp d’execution TribulleOpt en meilleur cas

2500	0.594	0.577	0.982	1.351	1.231	0.947
5000	2.447	2.465	2.446	2.561	2.453	2.4744
7500	3.868	3.928	4.183	3.743	4.413	4.027
10000	5.03	5.441	4.994	5.534	5.299	5.2596
12500	6.697	6.68	6.307	6.328	6.62	6.5264
15000	7.839	10.558	7.174	6.212	7.086	7.7738
17500	18.444	35.000999	20.419001	10.496	10.885	19.049
20000	20.650999	19.252001	17.618999	10.863	7.826	15.2421998
22500	11.927	12.024	11.423	11.312	10.708	11.4788
25000	12.027	11.97	11.517	11.921	12.017	11.8904
27500	12.862	13.656	12.805	13.333	13.271	13.1854
30000	13.95	14.24	14.194	14	14.204	14.1176
32500	16.059	16.645	16.377001	15.487	15.389	15.9914002
35000	28.813	22.099001	20.99	20.780001	20.941	22.7246004
37500	22.194	23.138	22.98	22.551001	22.388	22.6502002
40000	24.247	24.240999	24.562	24.167999	25.087999	24.4611994
42500	25.655001	26.118999	25.607	25.684	26.665001	25.9460002
45000	27.216999	28.468	27.667999	27.413	24.937	27.1405996
47500	26.150999	25.546	25.35	25.169001	25.329	25.509
50000	26.742001	26.499001	26.555	27.313999	27.716999	26.9654
52500	28.459999	28.367001	28.042	28.434	27.719999	28.2045998
55000	29.391001	29.266001	29.306	29.179001	29.281	29.2846006
57500	30.952	30.742001	30.992001	30.740999	30.451	30.7756002
60000	31.922001	31.809	32.076	32.337002	31.863001	32.0014008
62500	33.298	34.318001	33.084	33.443001	33.159	33.4604004
65000	34.409	34.393002	34.429001	34.793999	34.564999	34.5180002
67500	36.723	35.707001	35.838001	36.158001	35.949001	36.0750008
70000	37.715	37.235001	37.421001	37.241001	38.125	37.5474006
72500	38.465	38.495998	38.570999	38.516998	38.634998	38.5367986
75000	40.374001	39.825001	40.487	39.729	39.675999	40.0182002
77500	43.370998	41.087002	41.263	41.577999	41.654999	41.7907996
80000	42.140999	43.131001	42.070999	42.108002	42.193001	42.3288004
82500	44.742001	44.464001	43.814999	43.983002	44.361	44.2730006
85000	45.264	45.293999	45.318001	46.007	45.181	45.4128
87500	46.334999	46.727001	46.381001	46.928001	46.290001	46.5322006
90000	47.333	48.171001	47.463001	48.308998	47.402	47.7356
92500	50.910999	49.026001	45.660999	52.318001	46.216	48.8264



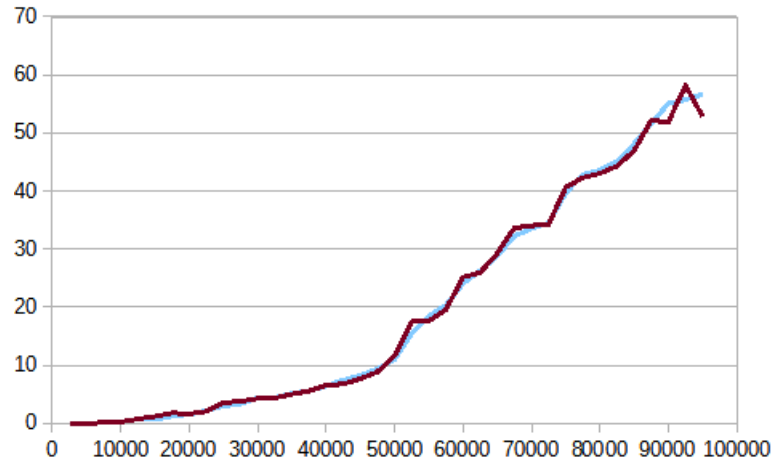
Le temp d’execution TribulleOpt en pire cas

2500	0.01	0.012	0.01	0.011	0.011	0.0108
5000	0.043	0.042	0.044	0.044	0.043	0.0432
7500	0.096	0.095	0.096	0.096	0.095	0.0956
10000	0.17	0.17	0.17	0.345	0.364	0.2438
12500	0.577	0.566	0.569	0.576	0.521	0.5618
15000	0.383	0.383	0.391	0.383	0.383	0.3846
17500	0.779	1.16	1.127	0.878	0.52	0.8928
20000	0.684	0.681	1.256	1.48	1.187	1.0576
22500	0.872	0.873	1.614	1.914	0.935	1.2416
25000	1.069	1.818	2.146	1.072	1.078	1.4366
27500	2.746	1.682	1.297	2.9	2.941	2.3132
30000	2.808	2.803	2.347	3.423	2.272	2.7306
32500	4.515	4.048	4.026	3.858	3.885	4.0664
35000	4.115	3.708	3.18	3.183	3.885	3.6142
37500	3.135	3.815	3.926	4.061	4.199	3.8272
40000	6.034	5.875	6.052	6.103	5.937	6.0002
42500	7.936	6.769	6.731	6.666	5.169	6.6542
45000	5.303	5.82	6.134	7.414	6.121	6.1584
47500	5.79	8.348	8.264	6.947	7.026	7.275
50000	6.569	9.104	6.646	7.475	7.055	7.3698
52500	7.445	6.563	6.553	6.945	7.526	7.0064
55000	8.604	9.132	10.889	8.712	7.188	8.905
57500	8.546	10.154	8.279	8.942	7.552	8.6946
60000	9.861	12.48	10.057	10.518	11.308	10.8448
62500	11.035	10.723	10.101	10.424	9.772	10.411
65000	10.363	10.388	10.899	10.944	10.955	10.7098
67500	12.016	15.186	11.575	11.485	11.507	12.3538
70000	12.461	12.896	12.831	12.075	12.062	12.465
72500	13.167	14.051	12.704	13.121	14.101	13.4288
75000	13.39	15.152	13.414	15.179	13.349	14.0968
77500	15.934	15.379	14.642	15.826	22.65	16.8862
80000	26.381001	27.811001	29.282	27.537001	19.125	26.0272006
82500	17.261	17.299999	23.080999	29.591	29.605	23.3675996
85000	33.254002	32.615002	33.118999	32.723	31.844999	32.7112004
87500	34.935001	33.383999	34.816002	34.728001	34.224998	34.4176002
90000	34.231998	36.433998	35.848	37.140999	34.540001	35.6389992
92500	38.041	37.650002	38.858002	37.896	36.776001	37.844201
95000	40.908001	40.386002	40.397999	40.655998	39.983002	40.4662004
97500	42.780998	43.609001	42.148998	41.782001	42.566002	42.5774
100000	44.890999	44.666	45.453999	44.675999	45.098999	44.9571992



Le temp d’execution TribulleOpt en moyen

2500	0.01	0.012	0.012	0.01	0.011	0.011
5000	0.05	0.05	0.051	0.05	0.05	0.0502
7500	0.125	0.123	0.124	0.125	0.125	0.1244
10000	0.233	0.232	0.247	0.271	0.235	0.2436
12500	0.509	0.828	0.833	0.803	0.774	0.7494
15000	0.56	0.561	0.563	0.586	1.207	0.6954
17500	1.801	1.076	0.797	0.786	1.802	1.2524
20000	1.903	1.027	1.104	2.231	1.63	1.579
22500	1.406	2.509	2.089	2.847	2.066	2.1834
25000	4.248	1.866	2.957	2.465	3.565	3.0202
27500	2.191	4.577	3.508	2.597	3.808	3.3362
30000	4.663	5.644	2.792	4.828	4.275	4.4404
32500	3.882	5.091	3.657	4.9	4.366	4.3792
35000	5.21	5.181	5.168	5.16	5.035	5.1508
37500	5.656	5.586	5.633	5.63	5.582	5.6174
40000	6.591	6.286	6.371	6.278	6.599	6.425
42500	8.076	6.997	7.74	7.66	6.776	7.4498
45000	8.371	8.479	7.956	8.952	7.677	8.287
47500	9.681	9.505	8.655	10.005	8.869	9.343
50000	11.116	9.638	10.448	12.33	11.627	11.0318
52500	13.208	13.587	16.922001	16.146999	17.513	15.4754
55000	18.992001	16.922001	19.559999	19.558001	17.594	18.5252004
57500	20.055	20.027	21.142	20.804001	19.577999	20.3212
60000	23.118999	23.947001	24.973	23.17	25.169001	24.0756002
62500	26.545	28.046	25.643	25.264	25.961	26.2918
65000	28.591	29.862	28.283001	27.913	29.257	28.7812002
67500	30.938	30.354	32.436001	33.845001	33.637001	32.2420006
70000	33.748001	34.144001	32.183998	33.375999	34.048	33.4999998
72500	33.846001	36.016998	34.210999	34.091	34.264999	34.4859994
75000	39.258999	40.676998	38.226002	39.560001	40.613998	39.6671996
77500	42.016998	44.411999	42.889999	42.43	42.308998	42.8115988
80000	42.686001	43.851002	42.234001	46.207001	43.014	43.598401
82500	47.074001	45.428001	44.445999	43.999001	44.291	45.0476004
85000	46.601002	45.702999	52.867001	48.462002	46.932999	48.1132006
87500	48.444	51.695999	57.606998	48.292	52.171001	51.6419996
90000	57.741001	52.880001	56.724998	56.042	51.855	55.0486
92500	52.479	53.970001	55.195999	58.771	58.179001	55.7190002
95000	67.053001	52.846001	60.793999	49.842999	52.84	56.6752
97500	47.251999	47.429001	49.155998	52.577	51.972	49.6771996
100000	56.584999	52.417999	47.688	48.778	47.807999	50.6553994



CONCLUSION

Nous pouvons conclure à partir des graphiques que la complexité du tri à bulles est en moyenne et pire O(n^2) pendant ce temps au mieux son O(n) car la fonction de tri ne fera qu'une boucle sur le tableau, lors de la comparaison du tribullopt et tribull, nous pouvons voir même s'ils partagent le même O(n^2) dans un cas moyen et pire et O(n) dans le meilleur des cas, que le tribullopt est plus rapide que le tribull.

L'idée de base derrière le tri à bulles est d'utiliser une variable d'indicateur pour savoir si des permutations ont été effectuées lors d'un passage dans le tableau. Si aucun échange n'a été effectué, cela signifie que le tableau est déjà trié et que l'algorithme peut se terminer plus tôt. Cela peut réduire considérablement le nombre d'itérations nécessaires pour trier le tableau, en particulier lorsque l'entrée est déjà partiellement triée.
La complexité temporelle du tri à bulles optimisé dans le pire des cas est O(n^2) , où n est le nombre d'éléments dans le tableau. Cependant, il fonctionne mieux que le tri à bulles ordinaire dans la pratique car il a un petit facteur constant et s'adapte aux cas déjà triés. La complexité temporelle optimale du tri à bulles optimisé est O(n) et elle se produit lorsque le tableau d'entrée est déjà trié.

À chaque passage dans le tableau, le plus grand élément « monte » jusqu'à la dernière position du tableau. C'est pourquoi on l'appelle "le tri à bulles". Ainsi, après chaque passage, le dernier élément est le plus grand du tableau, et il n'est pas nécessaire de le vérifier à nouveau, c'est pourquoi le tri à bulles optimisé réduit la taille du tableau à trier de un.

La complexité temporelle du tri à bulles opt dans le pire des cas est O(n^2), où n est le nombre d'éléments dans le tableau. Cependant, il fonctionne mieux que le tri à bulles classique dans la pratique car il s'adapte aux cas déjà triés. Comme le tri à bulles optimisé, la complexité temporelle dans le meilleur des cas est O(n) et elle se produit lorsque le tableau d'entrée est déjà trié.

Tri Gnome

le programme en C

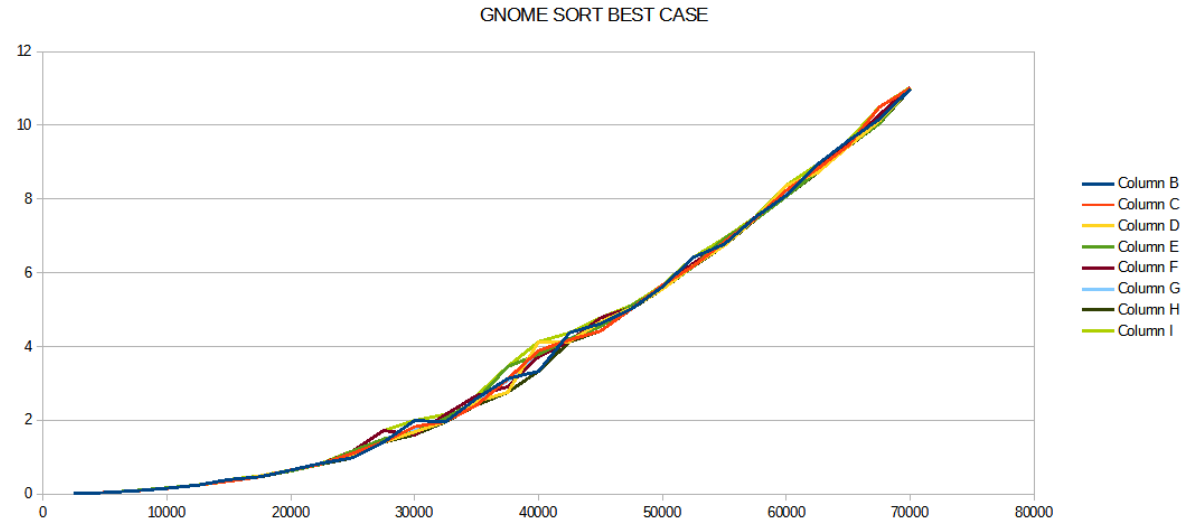
```
void gnomesort(int T[],int n)
{
  int i=0;
  int x;
  while(i<n)
  {
    if(i==0)
    {
      i++;
    }

    if(T[i]>=T[i-1])
    {
      i++;
    }

    else
    {

      x=T[i];
      T[i]=T[i-1];
      T[i-1]=x;

      i--;
    }
  }
}
```

Complexity theorique au meilleur cas

Complexity theorique au pire cas

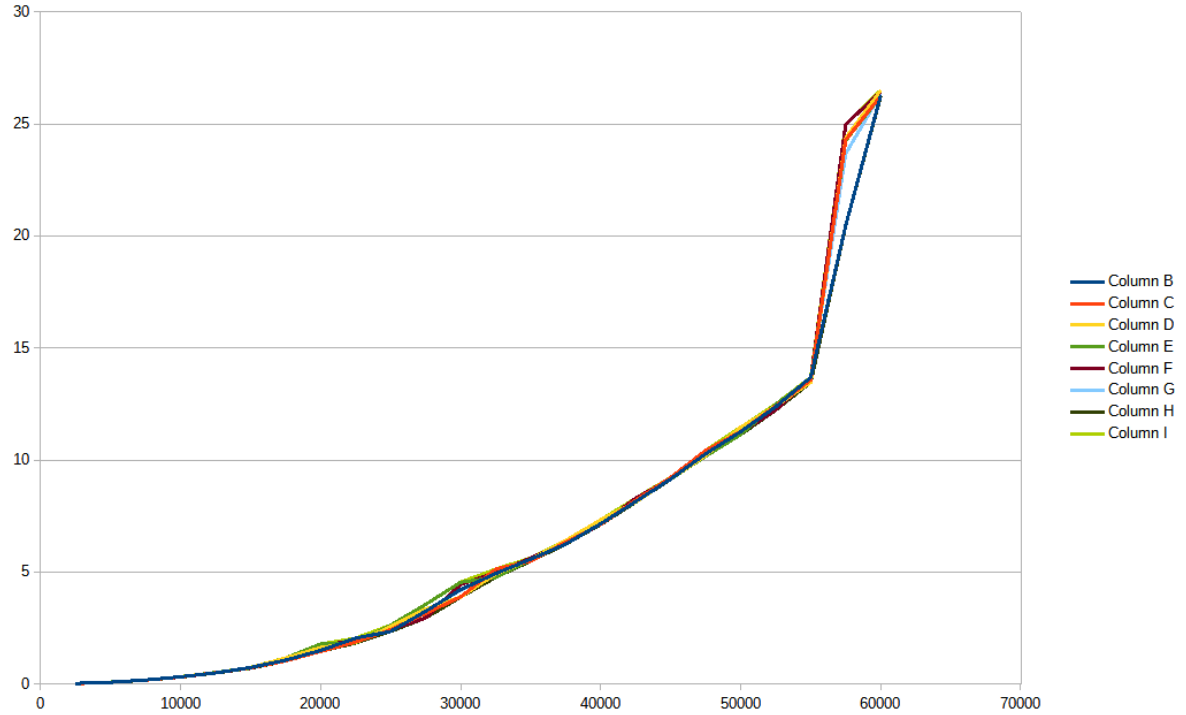
Le temp d’execution Gnome en meilleur cas

2500	0.01	0.011	0.01	0.011	0.009	0.0102	0.009	0.011
5000	0.04	0.039	0.039	0.039	0.039	0.0392	0.039	0.04
7500	0.087	0.086	0.086	0.094	0.094	0.0894	0.086	0.094
10000	0.155	0.153	0.154	0.167	0.154	0.1566	0.153	0.167
12500	0.24	0.242	0.241	0.24	0.242	0.241	0.24	0.242
15000	0.391	0.349	0.348	0.351	0.362	0.3602	0.348	0.391
17500	0.466	0.467	0.49	0.488	0.483	0.4788	0.466	0.49
20000	0.648	0.649	0.644	0.627	0.637	0.641	0.627	0.649
22500	0.838	0.821	0.836	0.809	0.816	0.824	0.809	0.838
25000	0.984	1.103	1.046	1.166	1.168	1.0934	0.984	1.168
27500	1.409	1.413	1.406	1.492	1.722	1.4884	1.406	1.722
30000	1.993	1.825	1.685	1.678	1.607	1.7576	1.607	1.993
32500	1.959	1.969	1.971	2.057	2.159	2.023	1.959	2.159
35000	2.592	2.411	2.509	2.543	2.668	2.5446	2.411	2.668
37500	3.127	3.116	2.754	3.452	2.908	3.0714	2.754	3.452
40000	3.33	3.889	4.131	3.791	3.728	3.7738	3.33	4.131
42500	4.372	4.182	4.13	4.208	4.148	4.208	4.13	4.372
45000	4.622	4.421	4.672	4.541	4.777	4.6066	4.421	4.777
47500	5.022	5.027	5.041	5.117	5.069	5.0552	5.022	5.117
50000	5.622	5.665	5.557	5.571	5.607	5.6044	5.557	5.665
52500	6.42	6.185	6.196	6.172	6.259	6.2464	6.172	6.42
55000	6.778	6.843	6.741	6.938	6.819	6.8238	6.741	6.938
57500	7.509	7.47	7.473	7.444	7.516	7.4824	7.444	7.516
60000	8.108	8.239	8.371	8.075	8.099	8.1784	8.075	8.371
62500	8.939	8.82	8.714	8.74	8.856	8.8138	8.714	8.939
65000	9.584	9.452	9.428	9.444	9.484	9.4784	9.428	9.584
67500	10.167	10.494	10.122	10.047	10.287	10.2234	10.047	10.494
70000	10.988	11.008	11.013	11.016	11.029	11.0108	10.988	11.029
72500	13.322	20.805	19.118	19.055	19.221001	18.3042002	13.322	20.805
75000	20.823	20.771999	20.622	20.636	20.594999	20.6895996	20.594999	20.823
77500	22.223	21.752001	22.247999	22.120001	21.924	22.0534002	21.752001	22.247999
80000	24.08	23.552999	23.971001	22.999001	23.393999	23.5994	22.999001	24.08
82500	24.822001	25.007999	20.117001	15.618	15.586	20.2302002	15.586	25.007999



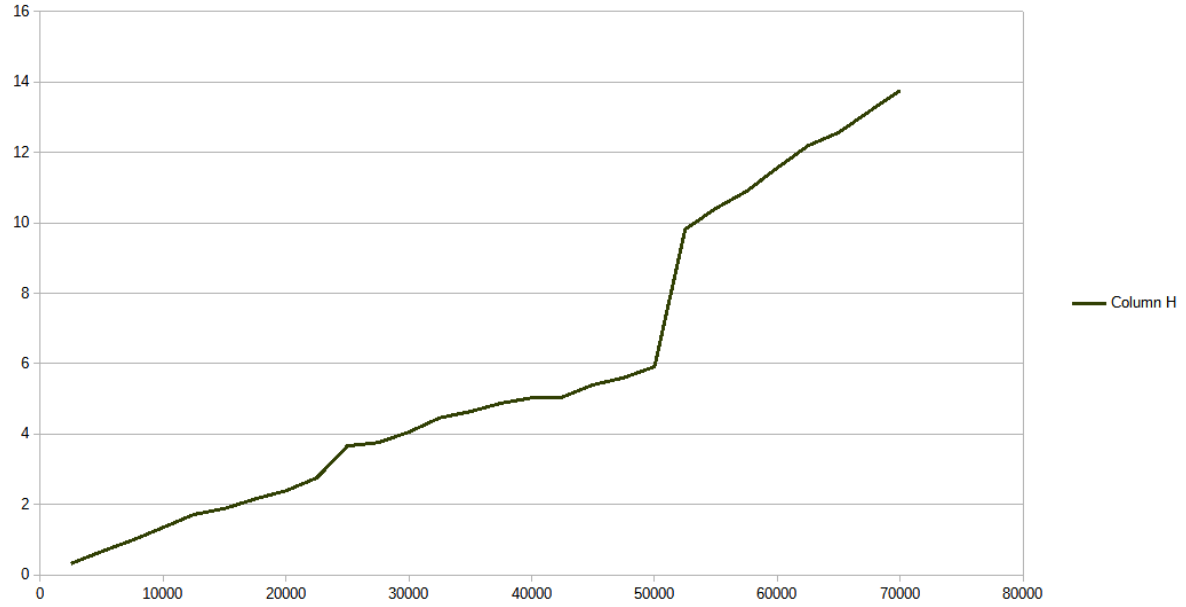
Le temp d’execution Gnome en pire cas

2500	0.026	0.02	0.02	0.02	0.02	0.0212	0.02	0.026
5000	0.08	0.081	0.078	0.082	0.078	0.0798	0.078	0.082
7500	0.177	0.173	0.178	0.178	0.177	0.1766	0.173	0.178
10000	0.324	0.32	0.323	0.317	0.313	0.3194	0.313	0.324
12500	0.495	0.501	0.502	0.514	0.503	0.503	0.495	0.514
15000	0.737	0.73	0.734	0.716	0.708	0.725	0.708	0.737
17500	1.066	1.029	1.158	1.163	1.128	1.1088	1.029	1.163
20000	1.5	1.456	1.628	1.786	1.761	1.6262	1.456	1.786
22500	2.046	1.877	1.852	1.835	1.908	1.9036	1.835	2.046
25000	2.346	2.425	2.56	2.624	2.423	2.4756	2.346	2.624
27500	3.247	3.15	3.346	3.553	2.951	3.2494	2.951	3.553
30000	4.214	3.887	3.88	4.541	4.448	4.194	3.88	4.541
32500	4.929	5.125	4.909	4.761	4.96	4.9368	4.761	5.125
35000	5.57	5.485	5.558	5.528	5.622	5.5526	5.485	5.622
37500	6.247	6.329	6.407	6.344	6.326	6.3306	6.247	6.407
40000	7.152	7.118	7.328	7.233	7.117	7.1896	7.117	7.328
42500	8.144	8.202	8.124	8.11	8.289	8.1738	8.11	8.289
45000	9.148	9.218	9.155	9.15	9.173	9.1688	9.148	9.218
47500	10.276	10.429	10.204	10.167	10.183	10.2518	10.167	10.429
50000	11.266	11.268	11.46	11.136	11.211	11.2682	11.136	11.46
52500	12.372	12.409	12.407	12.487	12.225	12.38	12.225	12.487
55000	13.69	13.569	13.468	13.624	13.637	13.5976	13.468	13.69
57500	20.469	24.285	24.374001	24.261999	24.979	23.6738	20.469	24.979
60000	26.278	26.25	26.497999	26.236	26.447001	26.3418	26.236	26.497999
62500	28.672001	28.283001	29.039	28.612	28.981001	28.7174006	28.283001	29.039
65000	31.179001	31.33	30.055	19.194	19.532	26.2580002	19.194	31.33
67500	20.812	20.503	20.493	20.570999	20.552999	20.5863996	20.493	20.812
70000	21.929001	18.695	20.214001	18.552	18.438	19.5656004	18.438	21.929001
72500	20.011999	20.082001	19.573999	19.929001	19.919001	19.9032002	19.573999	20.082001
75000	20.74	20.688	20.707001	20.978001	20.568001	20.7362006	20.568001	20.978001
77500	22.396	22.187	22.462999	22.315001	22.216	22.3154	22.187	22.462999
80000	23.658001	23.85	23.322001	28.622999	36.477001	27.1860004	23.322001	36.477001
82500	38.490002	39.514999	38.282001	39.834	39.943001	39.2128006	38.282001	39.943001
85000	40.729	41.123001	40.334	40.632999	41.127998	40.7893996	40.334	41.127998
87500	43.067001	43.028999	42.713001	43.131001	42.798	42.9476004	42.713001	43.131001
90000	45.941002	45.582001	45.167999	45.570999	44.419998	45.3363998	44.419998	45.941002
92500	42.471001	42.771999	43.539001	42.818001	42.918999	42.9038002	42.471001	43.539001
95000	45.008999	45.662998	45.151001	45.014999	45.848	45.3371994	45.008999	45.848
97500	47.466	48.674999	47.125	47.589001	32.321999	44.6353998	32.321999	48.674999
100000	31.236	31.223	31.223	31.145	31.136	31.1926	31.136	31.236



Le temp d’execution Gnome en moyen

2500	1.932	1.923	1.91	1.308	0.326	1.4798	0.326	1.932
5000	0.686	0.709	0.675	0.686	0.669	0.685	0.669	0.709
7500	1.059	0.991	0.989	1.03	1.038	1.0214	0.989	1.059
10000	1.395	1.435	1.414	1.35	1.626	1.444	1.35	1.626
12500	1.749	1.879	2.014	1.718	1.905	1.853	1.718	2.014
15000	2.116	2.518	1.888	1.912	1.901	2.067	1.888	2.518
17500	2.223	2.202	2.225	2.162	2.249	2.2122	2.162	2.249
20000	2.71	2.394	2.451	2.393	2.631	2.5158	2.393	2.71
22500	2.757	3.185	3.348	3.302	3.457	3.2098	2.757	3.457
25000	3.666	3.682	3.777	3.727	3.773	3.725	3.666	3.777
27500	3.836	3.76	3.979	3.963	3.859	3.8794	3.76	3.979
30000	4.117	4.17	4.059	4.133	4.194	4.1346	4.059	4.194
32500	4.76	4.579	4.463	4.71	4.785	4.6594	4.463	4.785
35000	4.641	5.148	5.05	4.899	4.959	4.9394	4.641	5.148
37500	4.879	5.084	5.253	5.258	5.283	5.1514	4.879	5.283
40000	5.637	5.321	5.03	5.419	5.627	5.4068	5.03	5.637
42500	5.878	5.55	5.14	5.055	5.274	5.3794	5.055	5.878
45000	5.602	5.523	5.401	5.707	6.36	5.7186	5.401	6.36
47500	7.889	6.806	5.603	5.703	5.683	6.3368	5.603	7.889
50000	5.915	6.278	16.087999	9.591	10.209	9.6161998	5.915	16.087999
52500	10.614	9.819	9.854	9.98	9.866	10.0266	9.819	10.614
55000	10.412	11.068	11.812	11.776	11.811	11.3758	10.412	11.812
57500	11.738	10.901	11.125	10.987	11.244	11.199	10.901	11.738
60000	11.563	12.194	12.545	11.561	11.75	11.9226	11.561	12.545
62500	12.193	12.397	12.966	12.496	12.493	12.509	12.193	12.966
65000	12.567	12.915	13.046	12.818	12.855	12.8402	12.567	13.046
67500	13.205	13.312	13.722	13.17	13.505	13.3828	13.17	13.722
70000	14.85	15.668	15.443	14.844	13.762	14.9134	13.762	15.668
72500	14.115	14.444	14.171	14.372	14.394	14.2992	14.115	14.444
75000	14.891	15	15.497	14.664	14.266	14.8636	14.266	15.497
77500	14.528	14.74	15.181	15.79	15.723	15.1924	14.528	15.79
80000	15.069	15.122	15.004	15.396	15.901	15.2984	15.004	15.901
82500	18.660999	16.872	16.08	15.341	15.917	16.5741998	15.341	18.660999
85000	16.794001	16.167	15.827	17.341999	17.483999	16.7227998	15.827	17.483999
87500	16.34	16.233999	16.538	16.451	16.599001	16.4324	16.233999	16.599001
90000	17.344	18.274	17.754999	17.104	16.737	17.4427998	16.737	18.274
92500	17.57	17.378	17.781	16.381001	12.079	16.2378002	12.079	17.781
95000	12.235	13.742	12.43	11.906	13.363	12.7352	11.906	13.742
97500	14.404	14.677	12.86	12.565	12.597	13.4206	12.565	14.677
100000	13.422	12.507	14.562	14.779	12.961	13.6462	12.507	14.779



CONCLUSION

Le tri Gnome est un algorithme de tri simple qui présente une complexité temporelle moyenne de $O(n^2)$ et dans le pire des cas de $O(n^2)$ où n est le nombre d'éléments à trier. Il utilise une comparaison en place pour trier les éléments adjacents. Même si sa complexité temporelle peut être inefficace en comparaison à d'autres algorithmes de tri, le tri Gnome est plus efficace que le tri à bulles en pratique.
Dans le meilleur des cas, lorsque les données sont déjà triées, le tri Gnome est très efficace avec une complexité temporelle de O(n), nécessitant seulement n-1 comparaisons et 0 échanges. Cependant, il est à noter que d'autres algorithmes de tri tels que le tri par insertion et le tri par sélection restent plus performants avec une complexité temporelle optimale de O(n) dans tous les cas.

La complexité temporelle de $O(n^2)$ dans le pire des cas pour le tri Gnome est due à son utilisation de la comparaison en place. Lorsque les données à trier sont dans un ordre inverse, cela signifie que le tri Gnome devra effectuer n-1 comparaisons pour chaque élément dans le tableau, soit un total de (n-1) * n / 2 comparaisons. Cela entraîne une complexité temporelle de $O(n^2)$ dans le pire des cas.

Tri par distribution

la fonction clé(E/ x, i : entier) : entier ;

```
int cle(int i, int j){

    int k1 = pow(10, j+1);
    int k2 = k1/10;

    int t = ((i % k1) - (i % k2))/k2;

    return t;
}
```

la fonction TriAux(T, n, i)

```
void triAux(int* T, int n, int i){
  // Step 1: Find the maximum digit of the current iteration
  int max = 0;
  for(int j = 0; j < n; j++){
    int digit = cle(T[j], i);
    if (digit > max) {
      max = digit;
    }
  }

  // Step 2: Initialize the count array and set all values to 0
  int count[max + 1];
  memset(count, 0, sizeof count);

  // Step 3: Count the occurrences of each digit
  for(int j = 0; j < n; j++){
    int digit = cle(T[j], i);
    count[digit]++;
  }

  // Step 4: Accumulate the count of each digit
  for(int j = 1; j <= max; j++) count[j] += count[j - 1];

  // Step 5: Place each element in its correct position in the output array
  int output[n];
  for(int j = n - 1; j >= 0; j--){
```

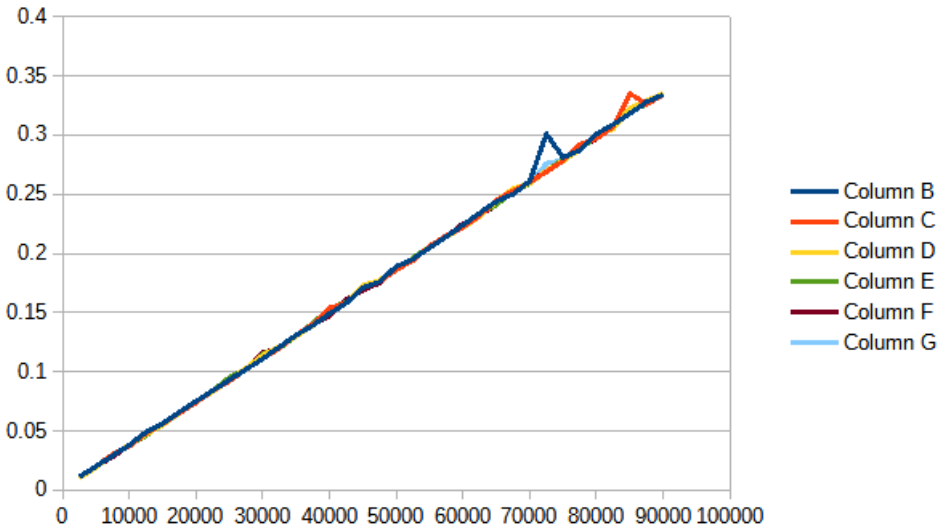
```
    int digit = cle(T[j], i);
    if(digit >= 0) {
      output[--count[digit]] = T[j];
    }
    else {
      output[--count[0]] = T[j];
    }
  }

  // Step 6: Copy the sorted elements back to the input array
  for(int j = 0; j < n; j++) T[j] = output[j];
}
```

la fonction TriBase(T, n, k)

```
void TriBase(int* T, int n, int k){
    for(int i=0; i<k; i++){
        triAux(T, n, i);
    }

}
```

Le temp d’execution Tri par distributionen en moyen

2500	0.011	0.011	0.01	0.011	0.01	0.0106
5000	0.02	0.02	0.019	0.02	0.02	0.0198
7500	0.029	0.03	0.03	0.03	0.028	0.0294
10000	0.038	0.037	0.038	0.038	0.038	0.0378
12500	0.049	0.048	0.047	0.046	0.047	0.0474
15000	0.056	0.056	0.055	0.056	0.056	0.0558
17500	0.066	0.065	0.065	0.066	0.065	0.0654
20000	0.075	0.074	0.075	0.075	0.075	0.0748
22500	0.084	0.084	0.083	0.084	0.084	0.0838
25000	0.093	0.092	0.093	0.095	0.094	0.0934
27500	0.102	0.102	0.103	0.102	0.102	0.1022
30000	0.111	0.111	0.114	0.115	0.116	0.1134
32500	0.121	0.12	0.121	0.12	0.12	0.1204
35000	0.131	0.131	0.13	0.13	0.13	0.1304
37500	0.139	0.14	0.139	0.141	0.14	0.1398
40000	0.149	0.154	0.152	0.153	0.147	0.151
42500	0.158	0.158	0.158	0.159	0.161	0.1588
45000	0.171	0.171	0.173	0.172	0.169	0.1712
47500	0.176	0.176	0.177	0.176	0.175	0.176
50000	0.189	0.186	0.188	0.186	0.189	0.1876
52500	0.195	0.194	0.195	0.196	0.195	0.195
55000	0.205	0.206	0.206	0.206	0.205	0.2056
57500	0.214	0.215	0.214	0.214	0.214	0.2142
60000	0.224	0.222	0.222	0.224	0.225	0.2234
62500	0.234	0.232	0.231	0.233	0.232	0.2324
65000	0.244	0.245	0.244	0.241	0.241	0.243
67500	0.25	0.253	0.255	0.252	0.253	0.2526
70000	0.261	0.26	0.259	0.259	0.26	0.2598
72500	0.301	0.269	0.27	0.27	0.27	0.276
75000	0.281	0.278	0.278	0.279	0.279	0.279
77500	0.287	0.292	0.288	0.288	0.289	0.2888
80000	0.301	0.297	0.3	0.299	0.297	0.2988
82500	0.309	0.307	0.305	0.306	0.307	0.3068
85000	0.318	0.335	0.323	0.32	0.319	0.323
87500	0.328	0.326	0.329	0.326	0.328	0.3274
90000	0.334	0.334	0.335	0.334	0.335	0.3344
92500	0.344	0.346	0.16	0.161	0.159	0.234
95000	0.167	0.165	0.165	0.164	0.166	0.1654
97500	0.173	0.168	0.169	0.169	0.17	0.1698
100000	0.242	0.375	0.372	0.372	0.373	0.3468



CONCLUSION

Le tri Radix est un algorithme de tri non basé sur la comparaison qui trie les nombres en regroupant les chiffres des nombres selon leur importance. Il est efficace pour les grands ensembles de données

contenant de nombreux nombres entiers, et est généralement plus rapide que les algorithmes de tri basés sur la comparaison tels que le tri rapide et le tri par fusion.

La complexité temporelle du tri Radix est O(d*n*), où d est le nombre de chiffres dans le plus grand nombre, n est le nombre d'éléments à trier.

Dans les meilleurs et pires cas, la complexité temporelle du tri Radix est la même, soit O(d*n). Cela en fait un algorithme de tri stable avec des performances constantes indépendamment des données d'entrée.