

[Sparse matrix multiplication – self code]

```

1  #include <cstdio>
2  #include <cstdlib>
3  #include <vector>
4  #include <utility>
5  #include <map>
6  #include <algorithm>
7
8  using namespace std;
9
10 const int TIMES = 1e7;
11 const int LIMIT = 1e6;
12
13 double av[TIMES];
14 int at[LIMIT];
15 int ac[LIMIT + 1];
16 int ar[TIMES];
17
18 double bv[TIMES];
19 int bt[LIMIT];
20 int bc[TIMES];
21 int br[LIMIT + 1];
22
23 map<int, double> m[LIMIT];
24 vector<pair<pair<int, int>, double>> vct;
25
26 int main() {
27     FILE *fa = fopen("ma.txt", "r");
28     FILE *fb = fopen("mb.txt", "r");
29
30     if (fa == NULL || fb == NULL) {
31         fprintf(stderr, "File ma.txt or mb.txt cannot open.\n");
32         exit(EXIT_FAILURE);
33     }
34
35     int x;
36     int y;
37     double v;
38
39     for (int i = 0; i < TIMES; i++) {
40         fscanf(fa, "%d%d%lf", &x, &y, &v);
41         ar[i] = x;
42         at[y]++;
43         av[i] = v;
44
45         fscanf(fb, "%d%d%lf", &x, &y, &v);
46         vct.push_back(pair<pair<int, int>, double>(pair<int, int>(x, y), v));
47     }
48
49     fclose(fa);
50     fclose(fb);
51
52     sort(vct.begin(), vct.end());
53     for (int i = 0; i < TIMES; i++) {
54         bt[vct[i].first.first]++;
55         bc[i] = vct[i].first.second;
56         bv[i] = vct[i].second;
57     }
58
59     for (int i = 0; i < LIMIT; i++) {
60         ac[i + 1] = ac[i] + at[i];
61         br[i + 1] = br[i] + bt[i];
62     }
63
64     for (int i = 0; i < LIMIT; i++) {
65         for (int j = ac[i]; j < ac[i + 1]; j++) {
66             int r = ar[j];
67             double va = av[j];
68
69             for (int k = br[i]; k < br[i + 1]; k++) {
70                 int c = bc[k];
71                 double vb = bv[k];
72             }
73         }
74     }
75 }

```

```

74         if (m[c].count(x) == 0) {
75             m[c][x] = va * vb;
76         }
77         else {
78             m[c][x] += va * vb;
79         }
80     }
81 }
82 }
83
84 FILE *of = fopen("own.txt", "w");
85 if (of == NULL) {
86     fprintf(stderr, "Output file cannot open.\n");
87     exit(EXIT_FAILURE);
88 }
89
90 for (int i = 0; i < LIMIT; i++) {
91     for (map<int, double>::iterator it = m[i].begin(); it != m[i].end(); it++) {
92         fprintf(of, "%d %d %.10f\n", it->first, i, it->second);
93     }
94 }
95 fclose(of);
96 }
97

```

```

b00902064@linux14:/tmp2/b00902064/z> time ./own
Total time spent: 11.648429
159.048u 4.828s 2:46.91 98.1%  0+0k 80+6234040io 0pf+0w

```

[Sparse matrix multiplication – matlab code]

```

1 function main()
2
3 [fid message] = fopen('ma.txt', 'r');
4 if (fid == -1)
5     disp(message);
6 end
7 ain = textscan(fid, '%f %f %f');
8
9 [fid message] = fopen('mb.txt', 'r');
10 if (fid == -1)
11     disp(message);
12 end
13 bin = textscan(fid, '%f %f %f');
14
15 t = cputime; tic;
16
17 for i = 1 : size(ain{1}):
18     ain{1}(i) = ain{1}(i) + 1;
19     ain{2}(i) = ain{2}(i) + 1;
20
21 for i = 1 : size(bin{1}):
22     bin{1}(i) = bin{1}(i) + 1;
23     bin{2}(i) = bin{2}(i) + 1;
24
25 sa = sparse(ain{1}, ain{2}, ain{3}, 1000000, 1000000);
26 sb = sparse(bin{1}, bin{2}, bin{3}, 1000000, 1000000);
27
28 ans = sa * sb;
29
30 fprintf('CPU time used %f\n', cputime - t);
31 fprintf('Elapsed time used: %f\n', toc);
32

```

```

>> main()
CPU time used 13.820000
Elapsed time used: 14.088336
>>

```