EE3980 Algorithms

HW5 Ranking Martial Artists

104061212 馮立俞

2018/4/8

Introduction

In this assignment, we are given 108 martial artist names and 63*5 = 315 1 vs. 1 match results. Then, we're required to rank the martial artists according to the matches.

TL;DR : See updates at the bottom of report.

Approach

The match result can be represented by a Directed Acyclic Graph(DAG) in which the edges point from match winner to match losers. Therefore, the graph consists of 108 vertices and 315 edges, which is too wasteful to construct the graph using adjacency matrix. As a result, I chose to build the graph using linked lists.

After the graph is constructed, we can sort the vertices using topology sort. It's noteworthy that since the graph is sparse, more than one valid sorting results are valid.

Topology Sort

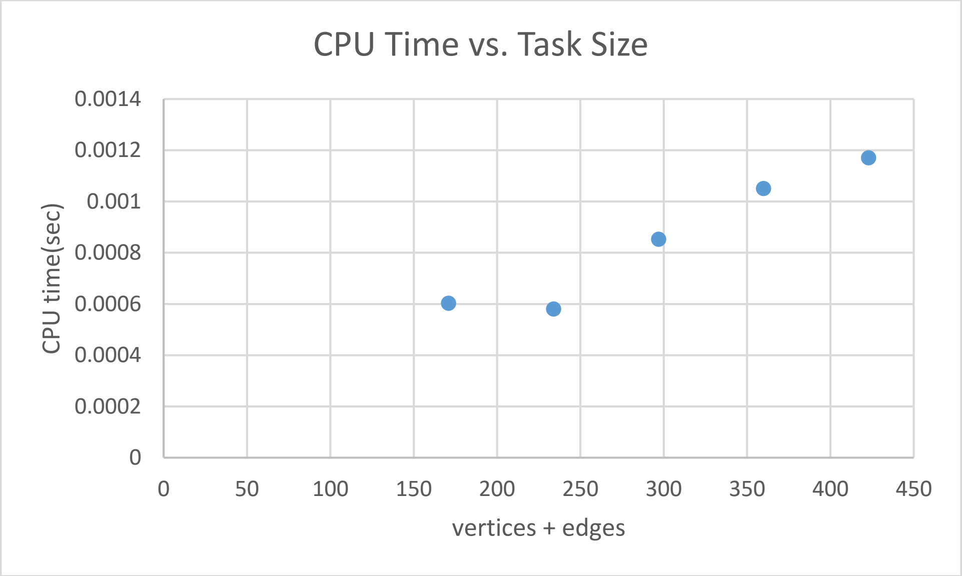
```
Algorithm top_sort(v, slist)
                  // Topological sort using depth first search algorithm.

// v is the vertex being visited; and slist is the ordered linked list.
{
        visited[v]: = 1;
        for each vertex w adjacent to v do {
            if (visited[w] = 0) then top_sort(w);
        }
        add v to the head of slist;

}
Algorithm topsort_Call(v) // Initialization and recursive top_sort function call.
 {

        for v: = 1 to n do visited[v]: = 0;
        slist: = NULL;

        for v: = 1 to n do
            if (visited[v] = 0) then top_sort(v, slist);

 }
```

In this algorithm, we use Depth First Search(DFS) to traverse the vertices. Since linked list is adopted, the complexity of traversal is $O(n+e)$, where n and e are the number of vertices and edges in the graph. If we use adjacency matrix, the complexity could grow to $O(n^2)$. Also, the space complexity of two data structure are $O(n+e)$ and $O(n^2)$, respectively.

Results and analysis

Efficiency

We can plot the overall execution time w.r.t. $n + e$ as follows



The curve is quite linear as e grows linearly. The abnormal behavior of the second point might be caused by initializing overhead.

Correctness

Since there’s more than one valid output, I didn’t check the output deterministically. Rather, I picked out some edges, then observed if the output obeys them. So far they’re valid.

Observations and Conclusion

Though being a linear complexity algorithm, the execution time of linked list approach seem to be much slower when it’s compared to other sorting algorithms in the previous assignments. This could result from the implemented data structure.

Update on 6/8:

File hw05.c was designed to handle all 5 given tournaments and be executed using below command.

```
cat player.txt tour1.txt tour2.txt tour3.txt tour4.txt tour5.txt | ./a.out
```

which I mentioned in submitted **usage.txt** file. Yet later I realized submitting more than 2 files is violation to the requirement mentioned in **hw05.pdf**. I'll attach the usage at the end of my report then.

Since I get the following comment for my last submission is,

[Program output] is incorrect.

And the correctness was verified by picking out a few martial artists to check their rankings. I wonder if the program is still waiting for 4 more input files to start sorting, thus showing nothing and was recognized as error.

To read in different number of files, one could edit the 117 line in **hw05.c**

```
117        for(j = 0; j < 5; j++){                    //tour1~tour5
```