Assignment 3

Nanyang Tang (z5103095)

**Question1:**

**(a).**

UC(T) = {a, b, c, d}

Reason:

e needs at least 3 steps to reach c.

f requires at least 3 steps to reach a.

g requires at least 3 steps to reach b.

**(b).**

TC(T) = {a, b, c, d, e, f, g}

Reason:

Since a, b, c, d, e, f, g could dominate each other.

**(c).**

CO(T) = {a, b, c}.

Reason:

a, c and b have the maximal outdegree which is 4.

**(d).**

BA(T) = {a, b, c}

Reason:

For a: acyclic subgraph of the tournament includes vertices: {a, c, d, g} with 6 edges.

For b: acyclic subgraph of the tournament includes vertices: {b, c, d, e} with 6 edges.

For c: acyclic subgraph of the tournament includes vertices: {c, a, d, f} with 6 edges.

For d: acyclic subgraph of the tournament includes vertices: {d, e, g} with 3 edges.

For e: acyclic subgraph of the tournament includes vertices: {e, a, f} with 3 edges.

For f: acyclic subgraph of the tournament includes vertices: {f, b, g} with 3 edges.

For g: acyclic subgraph of the tournament includes vertices: {c, e, g} with 3 edges.

**(e).**

the set of Condorcet winners is {}.

Reason:

All nodes have the parent node, therefore, there is no Condorcet winner.

**Question2:**

**(1).**

The Condorcet winner does not always have the maximum Borda score among all the alternatives.

For example,

|  |  |
| --- | --- |
| 6 voters |  |
| 3 voters |  |
| 4 voters |  |
| 4 voters |  |

Table 1

As can be seen from table1, a is preferred by the most voters. It is proved that

Compare a with b:

For a: the number of voters = 6 + 3 = 9

For b: the number of voters = 4 + 4 = 8. So a beats b.

Compare a with c:

For a: the number of voters = 6 + 4 = 10

For c: the number of voters = 3 + 4 = 7. So a beats c.

Therefore, a is the Condorcet winner.

Then we calculate the Borda scores.

For a: the scores = .

For b: the scores = .

For c: the scores = .

So, b is the Borda winner.

Consequently, the Condorcet winner does not always have the maximum Borda score among all the alternatives.

**(2).**

The Condorcet winner has at least half of the Borda score of the Borda winner.

Assume there are N voters and a is the Condorcet winner in m alternatives. This means that at least voters preferred a than other alternatives.

When a is the Borda winner, the scores of a is equal to the Borda score of the Borda winner.

When a is not Borda winner, we assume b is the Borda winner. In order to minimize the scores of a, we assume voters prefer the a and other voters rank a as the last alternative in all alternatives. As a result, the scores of a . Then, in order to maximize the scores of b, we assume we assume voters prefer b, and voters rank b as the second preferred alternative. This means the scores of b .

When is odd, and .

when

When .

when n.

Therefore, the Condorcet winner has at least half of the Borda score of the Borda winner.

**Question3:**

**(1).**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| agent | 1 | 2 | 3 | 4 | 5 |
| preferences |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Table 2

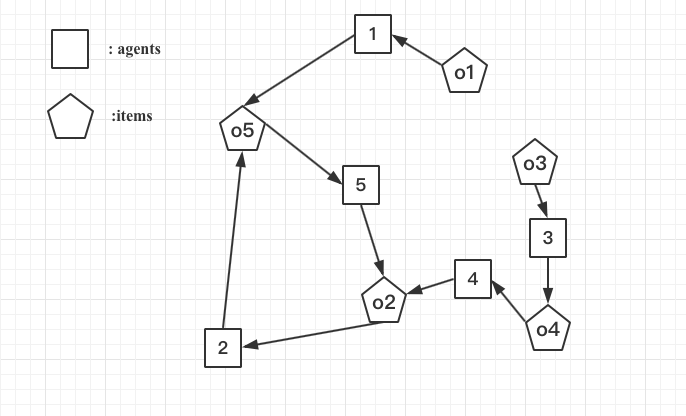


Figure 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| agent | 1 | 2 | 3 | 4 | 5 |
| preferences |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Table 3

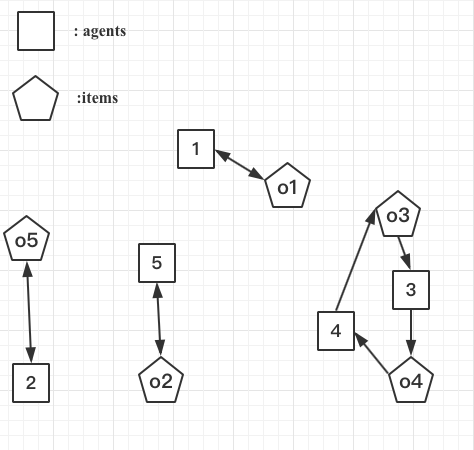


Figure 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| agent | 1 | 2 | 3 | 4 | 5 |
| preferences |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Table 4

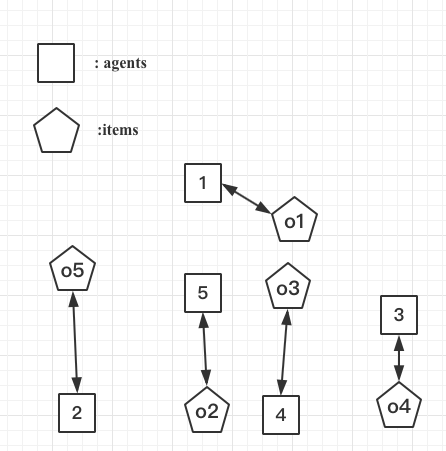


Figure 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| agent | 1 | 2 | 3 | 4 | 5 |
| preferences |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Table 5

The outcome of the TTC (top trading cycles) algorithm is shown in Figure3 and Table5.

**(2).**

If agent 4 misreports her preference, she could still not obtain a better a match. This is because and could be chosen preferably by 5 and 2. After that, agent 1 will choose . Therefore would become the best choice for agent4.

**(3).**

Therefore, the outcome is individually rational.

**Question 4**

**(1).**

1 proposes e, 2 proposes b, 3 and 4 propose a, 5 propose d.

a reject 3 in favor of 4 {{1, e}, {2, b}, {4, a}, {5, d}}

3 proposes b

b reject 2 in favor of 3 {{1, e}, {3, b}, {4, a}, {5, d}}

2 proposes a

a reject 4 in favor of 2 {{1, e}, {3, b}, {2, a}, {5, d}}

4 proposes b

b reject 4 in favor of 3

4 proposes c

{{1, e}, {2, a}, {3, b}, {4, c}, {5, d}}.

**(2).**

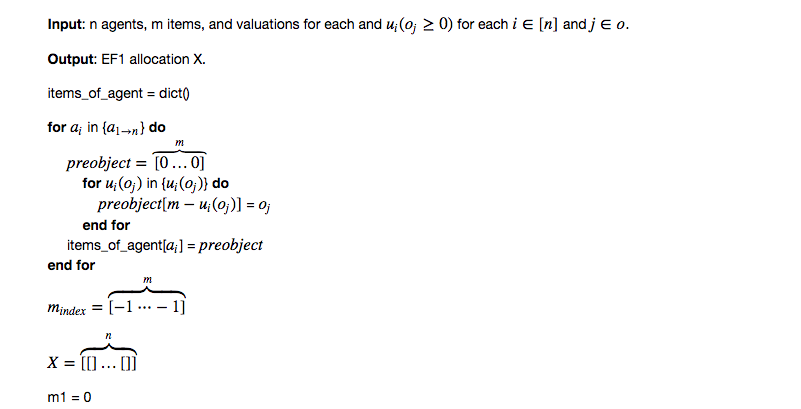
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | a | b | c | d | e |
| 1 | 2 | 0 | 1 | 3 | 4 |
| 2 | 3 | 4 | 2 | 1 | 0 |
| 3 | 4 | 3 | 2 | 1 | 0 |
| 4 | 4 | 3 | 2 | 1 | 0 |
| 5 | 1 | 3 | 2 | 4 | 0 |

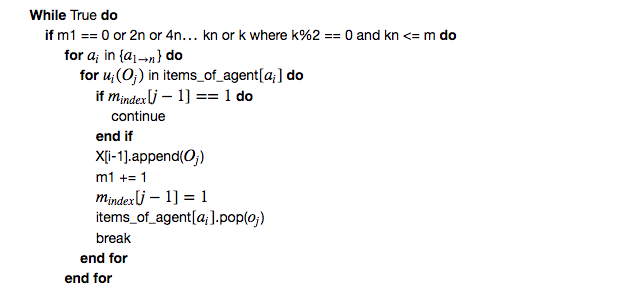
Table 6

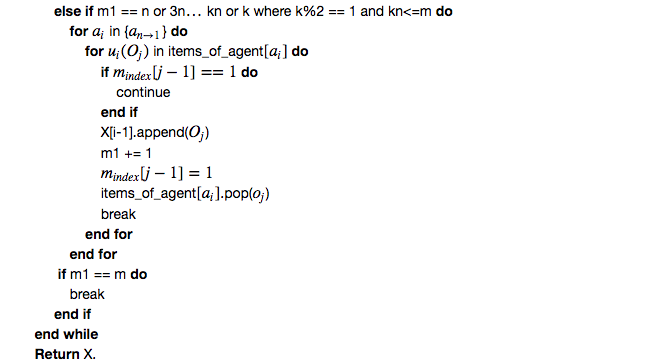
As can be seen from Table 6, the resultant matching is not Pareto optimal for students. Since for agent 2, is less than in terms of preference. This means there exists allocation Y such that for all and for some .

**Question 5**

**(1).**







**(2).**

As can be seen from the pseudocode above, firstly, we assume the utility of objects satisfies the straight preference. Next, according to the scores of different agents, we rearrange objects in decreasing preference. The time complexity is O(). Then, in the while loop, for each agent, she would only choose an item which has not been selected. In this way, at most n objects could be estimated. In addition, the round robin algorithm would be used in this process. In the whole process, and at most items would be chosen in accordance with the decreasing order of agents, and at most items would be chosen based on the increasing order of agents. Since there are m items assigned to n agents, and each agent would have n choices at worst case, the time complexity is . So, total time complexity is . Consequently, the time complexity is O(mn).

**(3).**

In this algorithm, each agent always chooses the preferential items in current state. Therefore, each agent could envy others due to the fact other agents choose one item that she prefers. But when we remove the item from other agents, she would not envy others’ allocation. Therefore, this algorithm always returns an EF1 allocation. Below, there is a worst example to explain it.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |

Table 7

As can be seen from Table 7, when the objects are sorted in accordance with the preference of different agents, agent 1 firstly chooses . After is selected, agent 2 must choose . Next, agent 3 only chooses . Now, agent 3 envies agent 1 and agent 2 and agent 2 envies agent 1, but they all fulfil envy-free 1. After the first round is completed, the selection priority-first of agents is opposite to the last round in this round. Consequently, for each agent, when an object is removed in the agent, no agent would envy it.

Therefore, this algorithm always returns an EF1 allocation.

**(4).**

Compared with the algorithm of Lipton et al. (2004), this algorithm would occupy more spaces. Each agent must have a list of objects in this algorithm, but the algorithm of Lipton only need a graph to save nodes. This means the algorithm of Lipton would reduce the space complexity.