Assignment 3

Answer – 1

(a) The uncovered set consists of {a,b,c,d}. Since, there exists a path from these nodes to every other node in at most two steps. E,f,g are not part of these sets because they cannot reach c,a,b respectively in at most two steps.

(b) The top cycle of tournament is {a,b,c,d,e,f,g} since every alternative has a path in the tournament to reach every other alternative.

(c) The Copeland winner is {a,b,c} since it has the maximal Copeland Score. This is justified because it has an outdegree of 4 which is the maximum in the tournament.

(d) The Banks winners are {a,b,c} since a is the top element of the maximal acyclic subgraph a,b,d,g. Similarly b in subgraph b,c,d,e and c in subgraph c,a,d,f. Also d has an acylcic subgraph d,f,g or d,e,f but it is not the maximal acyclic subgraph of the tournament as it only has a depth of 3 and two nodes below it unlike other winners. The rest of the alternatives do not form any acyclic subgraphs.

(e) Since, the number of individual voters and their selection is missing when we try to do a pairwise comparison we do not get a clear majority of alternative which is preferred over every other alternative. Thus, in this tournament there may not be a Condorcet winner.

Answer – 2

Since, Borda score is computed similar to a postional scoring rule and the theorem of Condorcet states that every PSR is not a condorcet-extension when there are 3 or more alternatives.
Also, the example below shows that a condorcet winner and a borda winner are not always the same-
where, A= {a,b,c}
6 voters: a>b>c
3 voters: c>a>b
4 voters: b>a>c
4 voters: b>c>a

In this case 'b' is the borda winner with a borda score of 5. While when do a paiwise majority of comparison 'a' is the winner. Thus making 'a' condorcet winner. Since, 'a' is preferred over 'b' by 9-8 and 'a' is preferred over 'c' by 10-7.

Thus, the above example shows that condorcet winner does not always have a maximum borda score among all the alternatives. Thus, disproving that the Condorcet winner always has the maximum Borda score among all the alternatives.

Although a the borda score of 'a' was 4 which was more than the half of the 'b' borda score. Hence, it shows that a condorcet winner has atleast of the borda score.
If we consider a worst-case scenario where -
10 voters: a>b>c
3 voters: c>b>a
4 voters: b>c>a

Still the borda winner remains 'b' with borda score of 4 while 'a' the condorcet winner has a borda score of 2.
This proves that the Condorcet winner has at least half of the Borda score of the Borda winner.

Answer -3

Here, each agents {1,2,3,4,5} owns a respective item {o_1,o_2,o_3,o_4,o_5} according to the endowment function.

Agent	1	2	3	4	5
preferences	o5	o5	o4	o2	o2
	o2	o4	o2	o1	o4
	o1	o3	o3	o5	o1
	o3	o1	o5	o3	o5
	o4	o2	o1	o4	o3

Initially the graph is computed from the preferences above where each item points to its owner and the owner points to its top most available item :-

o1->1->o5->5->o2->2->o5
o3->3->o4->4->o2->2

Here, we see there is a cycle in item5, agent5, item2 and agent 2, item5. So, the cycle is removed using TTC and we keep on processing till we get an empty graph.

After computing the TTC algorithm the outcome we get is that agent 1 owns item 1, agent 2 owns item 5, agent 3 owns item 4, agent 4 owns item 3 and agent 5 owns item 2.

Since, top trading cycle is strategyproof and agent 4 would not misreports her preferences or else she will be allocated any item less-preferred which would again coincide the strict preference rule. If she tried to change her preferences to get a better match works in the favour of the algorithm and she gets allocated accordingly which does not change the fact that the other agents also get their most preferred items.

Agent 4 already owns item 4 which makes it an individually rational scenario although the TTC algorithm ends successfully allocating her an item 3 which is her preferred item and less preferred by agent 3, the outcome is also individually rational.

Answer – 4

Each schools have only one seat. The student acceptance results as below -

1: e,b,a,c,d	a: 2,4,3,5,1
2: b,a,c,d,e	b: 3,2,4,5,1
3: a,b,c,d,e	c: 3,2,4,5,1
4: a,b,c,d,e	d: 5,2,4,3,1
5: d,b,c,a,e	e: 1,2,3,4,5

- 1 applies to e, 2 applies to b, 3&4 apply to a, 5 applies to d.
- a rejects 3 in favour of 4 {{1,e},{2,b},{5,d},{4,a}}.
- 3 applies to b.

- b rejects 2 in favour of 3 {{1,e},{3,b},{5,d},{4,a}}.
- 2 applies to a.
- a rejects 4 in favour of 2 {{1,e},{3,b},{5,d},{2,a}}.
- 4 applies to b.
- b rejects 4 in favour of 3 {{1,e},{3,b},{5,d},{2,a}}.
- 4 applies to c and gets accepted {{1,e},{3,b},{5,d},{2,a},{4,c}}.

The final outcome is that 1 is enrolled at e, 2 at a, 3 at b, 4 at c, 5 at d.

For pareto optimality we create the table below to according to the preferences of students only. The highest preference is given a utility value of 5 and then decreasing to 1 which would have to be the lowest preference. Here, the columns are the school names and the row are the students.

	A	B	C	D	E
1	3	4	2	1	⑤
2	4	⑤	③	2	1
3	⑤	4	③	2	1
4	⑤	4	③	2	1
5	2	4	③	5	1

So, the pareto optimal allocation would be given using the allocation formula u(X(i)) >= u(Y(i)).
X(1) = {e}, X(2) = {b,c}, X(3) = {a,c}, X(4) = {a,c}, X(5) = {d,c}.

Since, the pareto optimal solution matches with the final outcome of student 1,4 and 5. While for student 2 and 3 the allocating is a and b respectively but both of their allocation is not pareto optimal with respect to the final outcome calculated above.

Thus, we proved that the resultant matching is Not pareto optimal.

Answer – 5

The algorithm below is more likely to be faster than Lipton et al. (2004) which has a time complexity of O(n^3m) when we consider resource allocation.
The algorithm is –

```
1. List all items m to all users/agent n and keep them in the order of preference
2. While user u(i) of all users n has preferences
3. Do
    a. a(h) = first item on u's preference of list of items m, where 1≤h≤ m
    b. User u(i) requests item a(h)
    c. If  (some other user u(j) has item a(h)), where i≤j≤n
    d. Then
    e. Remove item a(h) from u(j)'s list
    f. For all items a(k), where h≤k≤ m
    g. Do
        i. Delete allocation a(k) of u(i+1) to u(n), here only the user u(i) with higher
           preference of a(k) gets that item and the others do not get that item and
           resulting in EF1 allocation.
4. Done
```

5. Print allocations of all users

Since there are n users the first computation takes a time complexity of O(n) and O(1) for that one item allocation. Thus, the total of that is O(n). Still there are m-1 computations more for the m-1 remaining items the complexity is then calculated for all m items resulting in the final time complexity of O(n) * O(m) = O(nm).

The advantage of Lipton et al. (2004) EF1 algorithm is there always exists an EF1 allocation and can be computed in polynomial time. While the algorithm above can compute the same only if such an EF1 allocation exists in shorter polynomial time.