EECS 336, Problem 7.3

Question a:

Reduction to maximum flow:

Algorithm 1 No Child Left Behind

Require: n, m, M

1: **function** NoChildLeftBehind(m, n, M)
2: Create vertices for each orphan and potential mentor, and add vertices s, s', t, t'.
3: **for** `each potential mentor j` **do**
4: Connect j to t with capacity 1.
5: **if** orphan i is in L_j **then**
6: connect direct edge $e = (i, j)$ with capacity 1.
7: **end if**
8: **end for**
9: **for** `each orphan i in the existing matching M` **do**
10: connect s' to i with capacity 1.
11: **end for**
12: connect t to s with capacity infinity (or a very large value).
13: Connect s to t' with capacity = "the number of orphans in the existing matching M" (Denote as a).
14: Let G_0 = current graph
15: **for** `each orphan i that is not in the existing matching M` **do**
16: Let $G' = G_0$
17: on G', connect s' to i with capacity 1.
18: c((s, t')) = a+1
19: connect s to all the orphans not in M except i
20: Taking s' as the source, t' as the sink, run Max-flow algorithm, max-flow = f*.
21: **if** f* = c((s, t')) **then**
22: Return True.
23: **end if**
24: **end for**
25: Return False
26: **end function**

RunTime:

Line 2 takes O(m+n+4) = O(m+n)

Line 3-8 takes O(m + $|L_i|$)

Line 9-13 takes O(n).

Line 14 takes O(m+n+$|L_i|$).

Line 15-25 can have at most n iterations, for each iteration, a max-flow takes O(n*($|L|+n+m+2$)). So totally it takes O($n^2 * (|L|+n+m)$). (Note $|L| \leq mn$)

Total Runtime O($n^2 * (|L| + n + m)$) or O($n^3 * m$).

Correctness of reduction:

The algorithm is actually created by two reductions. Here we will prove the correctness of both reductions individually.

Reduction 1:

We firstly add vertex s and t in addition to the n vertices for orphans and m vertices for mentors. Then the orphans and mentors are connected with capacity 1 based on L. We also connect all mentors to t with capacity 1. When we connect s to the orphans, 1) if orphan i is in M, then connect s to i by assigning both the lower and upper bounds as 1. 2) if orphan i in not in M, then connect s to i with capacity 1. 3) For testing if an orphan not in M can be matched with a mentor, we do iteration on all orphans i not in M individually, remove (s, i) and add (s', i) with capacity 1. Meanwhile add 1 to (s, t'). By following this process, we reduce our matching problem to a max-flow problem with lower bound (demand). The process here is identical to the reduction from bipartite matching to max-flow. And if there is a feasible solution to this network, this added orphan can be the one newly included.

Reduction 2:

Since the lower bounded capacity cannot be solved by max-flow algorithm, we have to convert (reduce) it further to a problem solvable by max-flow. We did the following steps:

1)Remove all edges with capacity [1,1].

2)Add edge (s',i) to all i's in M with capacity 1, add s to t' edge with capacity equal to the total number of existing matched orphans and add an infinity capacity edge (t,s) to compensate the lower bound requirement.

If the max flow on (s, t') equals to the number of edges with capacity [1,1] in Reduction1, then we call it a saturating flow. Then we show a feasible solution in Reduction 1 is equivalent to a saturating flow in Reduction 2. Let the lower bound on the edges in Reduction 1 be d(u,v), then step 1) and 2) are actually doing:
$f'(u,v) = f(u,v) - d(u,v)$
$f'(s',v) = \sum_{v \in V} d(u,v)$
$f'(u,t') = \sum_{w \in V} d(u,w)$
$f'(t,s) = |f|$
Since $f(e) \geq d(e)$, $f'(e) \geq 0$
Since $f(e) \leq c(e)$, $f'(e) \leq c'(e)$
And also $\sum_{u \in V'} f'(u,v) = \sum_{w \in V} f'(v,w)$, so f' is a legal (s',t') flow.The reduction is to reduce the flow by the lower bound and add the same amount to t'. The equivalency implies that for any saturated (s', t') flow f', the function $f = f'|_E + d$ is a feasible flow (s,t) in G.

Question b: No. Since we are testing if individual orphan that is not in M can be added to form a larger matching in each iteration, it is not guaranteed that other untested orphan cannot be matched. Therefore, it is not necessary to be a maximum matching.