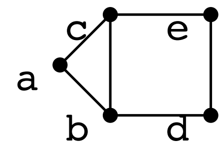
first name **last name** **id#**

each page 8 marks **30 min** **closed book** **no devices** **3 pages** **page 1**

1. Below is an integer program (IP) for finding a maximum indendent set in this graph. Also below is the dual of the IP.




```
primal                                  dual
max x1 + x2 + x3 + x4 + x5 s.t.        min y1 + y2 + y3 + y4 + y5 + y6 s.t.
    x1 + x2                <= 1            y1 + y2                     >= 1
    x1      + x3           <= 1            y1      + y3 + y4           >= 1
         x2 + x3           <= 1                 y2 + y3      + y5      >= 1
         x2      + x4      <= 1                          y4      + y6  >= 1
              x3      + x5 <= 1                               y5 + y6  >= 1
                   x4 + x5 <= 1
x1, x2, x3, x4, x5 in {0, 1}               y1, y2, y3, y4, y5, y6 >= 0
```

2. What does primal variable x_3 represent?

Answer: variable x_3 represents the decision that the node c be included in the independent set or not.

What does dual variable y_3 represent?

Answer: Dual linear program corresponds to the minimum edge cover problem in the graph above. So, variable y_3 represent the decision that the edge 'cb' in graph above be included in the edge cover or not.

Justify/explain the primal objective function:

Answer: primal objective computes the number of nodes in a feasible independent set and since we are trying to find the maximum independent set, we try to maximize this objective.

Justify/explain the dual objective function:

Answer: dual objective computes number of edges in a feasible edge cover for the graph.

Justify/explain this primal constraint: $x_1 + x_2 \leq 1$.

Answer: Since there is an edge between nodes a,b, then at most one of them could be included in an feasible independent set and the constraint ensures this (variables x_1 and x_2 correspond to nodes b and a).

Justify/explain this dual constraint: $y_1 + y_3 + y_4 \geq 1$.

Answer: this constraint ensures that the node b would be covered by a feasible edge cover for the graph above. Since y_1,y_2,y_3 correspond to edges incident to node b, then at least one of them should be selected by a feasible edge cover.

3. Is $x = (1, 0, 0, 0, 0)$ primal optimal? If yes, write it below: if no, find a primal optimal solution.

Is $y = (1, 1, 1, 1, 1, 1)$ dual optimal? If yes, write it below: if no, find a dual optimal solution.

Your primal optimal solution:

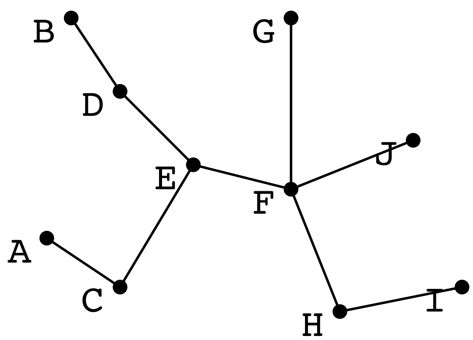
x=(1,0,0,0,1)

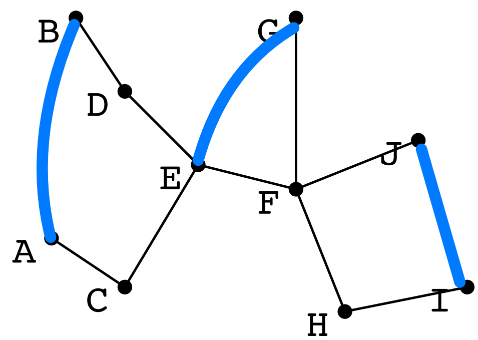
Your dual optimal solution:

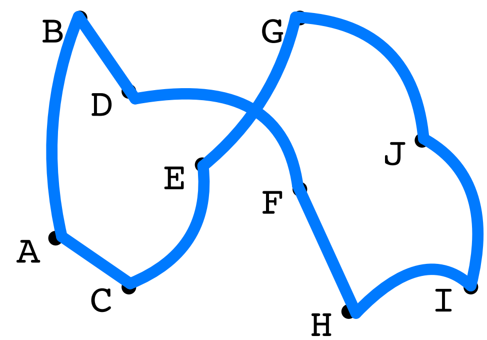
y=(1/3,1/3,1/3,0,0,1)

4. For the complete graph with these edge weights, give the Hamiltonian cycle found by the 1.5-approx method. When shortcutting your Eulerian tour, start at node E and whenever you have a choice, pick the alphabetically-first choice. Use below left for your rough work. Show your matching-plus-MST below middle. Show your cycle below right.

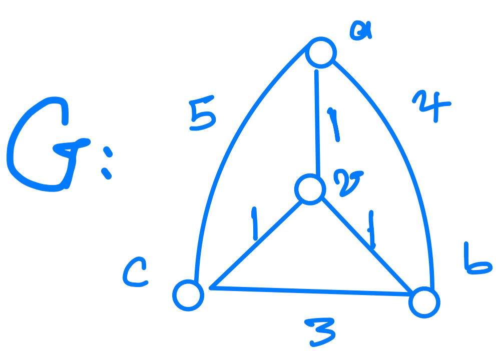
	B	C	D	E	F	G	H	I	J
A	5	2	4	4	6	7	7	9	9
B		6	2	4	6	5	9	11	8
C			4	3	4	7	5	8	7
D				2	4	4	7	9	7
E					2	4	5	7	5
F						4	3	4	3
G							7	7	4
H								3	4
I									4

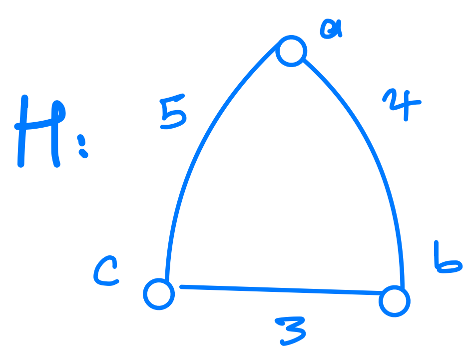






5. Let G be a weighted complete graph with at least 4 nodes and positive edge weights. Let v be a node in G and let H be the graph obtained by removing v and all edges incident with v. Let h be the weight of a min-weight Hamiltonian cycle in H. Let g be the weight of a min-weight Hamiltonian cycle in G. Prove/disprove: $h \leq g$.





Disprove. As can be seen above by deleting the node v from graph G the weight of min-weight cycle increases from 9 to 12 in graph H. (However, if we had triangular inequality between weights of the edges of the graph then the claim would be true, since we can shortcut over the vertex v in the min weight cycle of graph G and obtain a cycle of smaller weight in graph H.)

first name **last name** **id#**

each page 8 marks **30 min** **closed book** **no devices** **3 pages**

6. On this instance, find the greedy set cover: in each step, if there are ties then pick the set with smaller index, e.g. if there is a tie between picking S5 and S9, pick S5. Also, find a minimum size set cover. Write each cover like this: {S0, S1, S2, S4}.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S0	-	-	*	*	-	-	*	-	-	-	*	-	-	-	*	-
S1	-	-	*	*	-	-	*	-	*	-	*	*	-	*	*	-
S2	-	-	-	*	-	-	-	-	-	-	*	-	-	-	*	-
S3	-	-	*	-	*	-	-	*	-	-	*	-	-	-	*	*
S4	-	-	-	-	*	*	-	*	*	*	-	-	-	-	-	-
S5	-	-	*	-	*	-	*	-	-	*	-	*	*	-	-	-
S6	-	*	*	-	*	-	*	-	-	*	-	*	*	-	-	-
S7	*	-	-	*	-	*	-	*	*	-	-	-	-	*	-	*
S8	-	*	*	-	*	-	*	-	-	*	-	*	*	-	*	-
S9	*	-	-	*	-	*	-	*	*	-	*	-	-	*	-	*

a) **your greedy set cover:**

Answer: {S1, S6, S7}

b) **your min-size set cover:**

Answer: {S8,S9}

Rough work here

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S0	-	-	*	*	-	-	*	-	-	-	*	-	-	-	*	-
S1	-	-	*	*	-	-	*	-	*	-	*	*	-	*	*	-
S2	-	-	-	*	-	-	-	-	-	-	*	-	-	-	*	-
S3	-	-	*	-	*	-	-	*	-	-	*	-	-	-	*	*
S4	-	-	-	-	*	*	-	*	*	*	-	-	-	-	-	-
S5	-	-	*	-	*	-	*	-	-	*	-	*	*	-	-	-
S6	-	*	*	-	*	-	*	-	-	*	-	*	*	-	-	-
S7	*	-	-	*	-	*	-	*	*	-	-	-	-	*	-	*
S8	-	*	*	-	*	-	*	-	-	*	-	*	*	-	*	-
S9	*	-	-	*	-	*	-	*	*	-	*	-	-	*	-	*

7. Consider the greedy set cover algorithm on an instance with 50 elements and minimum cover size 9. Let n_t be the number of elements not yet in the cover after t iterations of the algorithm.

a) Give the best upper bound you can for n_1. Justify briefly.

The first set chosen by greedy algorithm is the set with maximum number of elements between all sets. Now the largest set should have size of at least 6 otherwise, no collection of 9 sets of size at most 5 can cover 50 elements because they can at most cover 45 elements. As a result, the size of largest set is at least 6 and the greedy algorithm by choosing that set will leave at most 44 number of elements uncovered.

a) Give the best upper bound you can for n_2. Justify briefly.

39. The greedy algorithm now in the second iteration selects the set that covers maximum number of uncovered elements by considering the items that already have been covered in the first iteration. From the last part, we know that there are most 44 number of uncovered elements after the first iteration. Now the size of the set that has maximum number of uncovered elements after the first iteration should be at least 5, as a result the number of uncovered elements after the second iteration would be at most 39. The size of set with maximum number of uncovered elements in second iteration is at least 5 since the minimum covering for the remaining 44 elements would have size at most 9 and no collection of 9 sets of size at most 4 can cover 44 elements because they can at most cover 36 elements.