Q1-10 True and False

1. [True] In the Ford-Fulkerson algorithm, the choice of augmenting paths can affect the number of iterations.

   Explanation: Choosing the shortest path can efficiently decrease the number of iterations

2. [True] If 3-SAT $\leq_p$ 2-SAT, then P = NP.

   Explanation: 3-SAT is in NPC, 2-SAT is in P, thus we show that all NP problems can be reduced to 2-SAT in polynomial time by first reduced to 3-SAT first, as the definition of NPC.

3. [True] Every problem in P can be reduced to 3-SAT.

   Explanation: Since P is a subset of NP and 3-SAT is a NPC problem, every problem in NP, including every problem in NP, can be reduced to 3-SAT polynomially.

4. [False] A linear program with all integer coefficients and constants must have an integer optimum solution.

   Explanation: It doesn't have to. Consider max y, subject to x >= 0, y >= 0 and x + 2y <= 1. The optimum solution is ½ where y = ½.

5. [False] The residual graph of a maximum flow f is strongly connected.

   Explanation: It should be disconnected.

6. [False] If problem Y is polynomial time reducible to problem X, then X is polynomial time reducible to Y.

   Explanation: Let Y be 2-SAT and X be 3-SAT. We don't know whether 3-SAT is reducible to 2-SAT.

7. [True] In a flow network G, if we increase the capacity of one edge by a positive amount x and we observe that the value of max flow also increases by x, then that edge must belong to every min-cut in G.

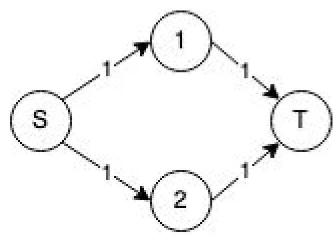
Explanation: Suppose there is one edge, e, and one min-cut, S, such that e does not belong to S, then we know that e is not the bottleneck. Increasing e's capacity will not help.

8. [False] If a linear program has an unbounded value, then its dual is also unbounded.

   Explanation: The dual should have no feasible solution.

9. [False] Suppose the max-flow of a graph has value f. Now we increase the capacity of every edge by 1. Then the max-flow in this modified graph has value at most f+1.

   Explanation: Consider the following graph. If we increase the capacity of all edge by 1, then the max-flow will increase from 2 to 4.



10. [False] If X is in NP and we can reduce a known NP-Complete problem to X, then X is NP-Complete.

    Explanation: It should be a polynomial reduction.

Q11-20 Multiple Choices

Q11

Which of the following statement(s) is true?

1. Every circulation is a flow
2. **In the Ford-Fulkerson algorithm, one can use either BFS or DFS to find an augmentation path.**
3. For every node in a network, the total flow into a node equals the total flow out of a node.
4. All of them are true

Explanation: 1 is false because a circulation does not have a source or terminal, while flow must have. 3 is false since the flow into does not equal to the flow out of the source. 2 is True because we can run BFS/DFS on the residue graph to find an augmentation path.

Q12

What would be the running time of the Ford-Fulkerson algorithm if we always choose the augmenting path with the least number of edges?

1. **$O(E^2 V)$**
2. $O(E V)$
3. $O(E V)$
4. $O(V^2 E)$

Explanation: This improved version of FF algorithm is Edmonds–Karp algorithm, whose complexity is $O(E^2 V)$.

Q13

What is the size of the maximum flow between two arbitrary vertices in the K_n (n>1) network such that all edges have capacity 1?

1. n
2. n+1
3. 1
4. **n-1**

Explanation: The max flow can be achieved from S to T by the following route: S → T, and S → v → T for all vertices that are neither S nor T. All route's capacities are 1 and there are n-1 routes. Thus the final answer is n-1.

Q14

What is the maximum number of edges a bipartite graph with 20 vertices can have?

1. 400
2. **100**
3. 80
4. 20

Explanation: It is the same to find the maximum value of x(20 - x) subject to 1 <= x <= 20. When x = 20 / 2 = 10, we can get the maximum value, which is 10 * (20 - 10) = 100.

Q15

Choose the statement that holds true for problems in the NP-Complete (NPC) class.
I) Can be solved in polynomial-time.
II) Some of them are optimization problems.
III) The solution of a NPC problem can be verified in polynomial time.
IV) There exists no proof that a polynomial-time algorithm cannot solve any NPC problem.

1. II & IV
2. I & II
3. II & III
4. **III & IV**

Explanation: We don't know whether I) is true unless P = NP. II) is false since optimization problems are not a decision problem. III) is true since we can find a verifier for all NP problems, including NPC problems, to verify the solution in polynomial time. IV) is true because it is equivalent to say there exists no proof of "P = NP".

Q16

What is the time complexity to find the optimal way of computing the matrix product of a sequence of n matrices so that the number of scalar multiplications is minimized?

1. $\Theta(n^2)$
2. $\Theta(n^2 \log n)$
3. **$\Theta(n^3)$**
4. $\Theta(n^{2.7\ldots})$

Explanation: It is a DP problem discussed in the lecture.

Q17

Let X be a problem that belongs to the class NP. Which one of the following is true?

1. X may be undecidable.
2. If X can be solved deterministically in polynomial time, then P = NP.
3. X cannot be solved deterministically in polynomial time.

4. **X can be solved in nondeterministic polynomial time.**

Explanation: 4) is true since it is the definition of NP. 1) is false since for a NP problem, we can find a verifier to verify the solution, which makes that problem decidable. 2) is false because X might not be an NPC problem. 3) is false because it implies P is not NP, which we do not know if that is true or false yet.

Q18

Consider the linear program:

max (31 x + 55 y)

subject to

x + y <= 80
x >= 2 y
x,y >= 0

Which of the following statement(s) are true?

I) the linear program has a single optimal solution
II) the feasible region is bounded
III) the linear program has infinitely many optimal solutions

1. III
2. None
3. II & III
4. **I & II**

Explanation: The optimal solution is when x = 2y and x + y=80. Thus, I) is ture and III is false. II) is true and can be verified by drawing the graph of all the constraints.

Q19

Let S be an NP-complete problem and Q and R be two other problems not known to be in NP. Q is polynomial time reducible to S and S is polynomial-time reducible to R. Which one of the following statements is true?

1. R is NP Complete
2. Q is NP Complete

3. **R is NP Hard**
4. Q is NP Hard

Explanation: As per the statement, we know that Q $\leq_p$ S $\leq_p$ R. Thus, only 3) is true.

Q20

Suppose that you run the Edmond-Karp algorithm to solve a bipartite matching problem on a $K_{n,n}$ graph. How many augmenting paths are needed in the worst case?

1. O(1)
2. $O(n^2)$
3. **O(n)**
4. O(n log n)

Explanation: Since the EK algorithm will always pick the shortest path, for each iteration, it will always pick the path connecting two nodes that has never been selected before. We can always find this path since it is a $K_{n,n}$ graph. All augmenting paths that pass the selected node will include backward edges and will not be the shortest path. Thus, since there are at most n matching, the time complexity is O(n).

Q21 Dynamic Programming

Q21.1

Answer: Let OPT[v] be the maximum weight of an independent set in the subtree rooted at node v.
Rubrics:
- -0.2 for missing “independent set” or “tree rooted at x”
- -1.2 for more vague formulation.
- -2.2 to -3.2 for the wrong ones.

Q21.2

Answer:
OPT[v] = max(w[v] + sum{OPT[u] for u in children of children[v]}, sum{OPT[u] for u in children[v]});
Base case: OPT[v] = w[v] for all v in leaves;

Rubrics:

2 points for the base case, 6 points for the recursion relation: 3 points for the case that includes the root, and 3 points for the case that doesn't include the root.

Additionally,

- -3 points if the students used 1 rather than w[v] in the recursion relation: that is the answer for the largest independent set
- -2 if the student assume the tree is a binary tree, but give the correct recursion relation and the base case
- -2 if the student never mentioned summation (sum, Σ, etc.) in the recursion relation

Q21.3

Answer: Fill up the one-dimensional table. The solution is OPT[r].

Rubrics:

- 3.2 marks if data structure and where to find the solution provided.
- 2.7 marks for explaining the bottom up approach, but missing data structure (array/table) details and also where to find the final answer (OPT[r]).
- 2.2 marks if bottom up approach explained using just pseudo code (not using Plain English)
- 1.5 mark if bottom up not explained and provided information on how to find the combination.
- 1 mark if incorrect algorithm provided

Q21.4

Answer: Linear with respect to the number of nodes in the tree. For each node, its value will be accessed at most twice: when we calculate OPT for its parent and its grandparent if exists.

Rubrics:
The students get:

- 1.6 point if correct
- 1 point if the explanation is wrong but the time complexity is correct
- 0.8 point if the time complexity is not optimal, but the explanation is correct
- 0 point if everything is wrong.

Q22 Linear Programming

Q22.1

Answer:
max (3 z1 + 8 z2 – z3 – 3 z4) subject to

$z1 + 4\ z2 + 2\ z3 - z4 \le 34;$

$- z1 - z2 + z3 + z4 \le -3;$

$z2 - z3 \le -1;$

$- z2 + z3 \le 1;$

$z1, z2, z3, z4 \ge 0;$

Rubrics:
1 point for each correct expression.

Q22.2

Answer:
min(34 y1 – 3 y2 – y3 + y4) subject to

$y1 - y2 \ge 3;$

$4y1 - y2 + y3 - y4 \ge 8;$

$2y1 + y2 - y3 + y4 \ge -1;$

$-y1 + y2 \ge -3;$

$y1, y2, y3 \ge 0;$

Rubrics:
- -1 point for each mistake made (wrong sign, operator, etc.)
- -2 if the object function is wrong
- -3 if the object function is missing

Q23 Circulation

Q23.1

Answer:
D(A)=-4, D(B)=6, D(C)=7, D(D)=-12, D(E)=2, D(F)=1
Rubrics:
1.25 for each answer, 7.5 in total

Q23.2

Answer:
Max flow = 16
Rubrics:
4.5 if the answer is correct, 0 otherwise

Q23.3

Answer:
It is feasible because the maxflow value equals the sum of demands.

Rubrics:
3 if the answer is correct, 0 otherwise

Q24 Max-flow

Q24.1

Answer: Two partitions: four projects and four months. They connected accordingly. P1 is connected to m2. P2 is connected to all months. P3 is connected to m1 and m2; P4 is connected to m2 and m3. The weight on all those edges is 5. Connect the source s with projects with weights 8, 10, 7 and 12 respectively. Connect months with the target by edges with weight 7.

Rubrics:
-1 for each mistake made (e.g. missing capacity, wrong/missing edge, no mentioning the source/sink, etc.)

Q24.2

Answer: To check whether all projects can be completed within the time limits, it suffices to check whether the network admits a feasible flow of value 8 + 4 + 7+ 3 = 22. Such a flow does exist.

Rubric:
3 points - Feasible flow exists or not (Misinterpretation).
3 points - Explanation of why flow exists with a feasible flow value or not based on the constructed graph.
-2 points - If feasible flow value isn't mentioned.

Q25 Weak-SAT

Q25.1

Answer: It is in NP as given an assignment, we can check whether each clause is satisfied. If there are exact (m-1) satisfied clauses, we know that it is a feasible solution and otherwise, it is not. These operations can be done in polynomial time.

Rubrics: 1.8 if the solution mentioned a polynomial time verifier with details. 1 point if the time complexity is wrong and it doesn't say polynomial time. 0 if there is no verification or the verification step is totally wrong.

Q25.2

Answer: Given a CNF formula F. Add two clauses (x1) ∧ (nx1) to it creating a new formula F' = F ∧ (x1) ∧ (nx1).

Rubrics: 9 points if the reduction is correct. 2 points if the reduction is not correct but contains some good ideas. 0 point if it is not attempted.

Q25.3

Answer: Claim: F is satisfiable if and only if F' is weak-satisfiable.,
Rubrics: 1.8 points if claim is correct. 0 point if the above reduction is wrong or the claim is wrong.

Q25.4

Answer: If F is satisfiable, then one of these new clauses (x1) or (nx1) is satisfied. So F' is weak-satisfiable.
Rubrics:

- 2.7 If the student's solution is consistent with the published solution or the student's solution works using its reduction in Q25.2.
- 0 if the student's solution does not work at all.

Q25.5

Published solution:

Construction: Given a CNF formula F. Add two clauses (x1) ∧ (nx1) to it creating a new formula F' = F ∧ (x1) ∧ (nx1).,
Claim: F is satisfiable if and only if F' is weak-satisfiable.,
From SAT to the reduced problem: If F is satisfiable, then one of these new clauses (x1) or (nx1) is satisfied. So F' is weak-satisfiable.,
From the reduced problem to SAT: If F' is weak-satisfiable. The only unsatisfied clause must be one of (x1) or (nx1), so all the original m clauses are satisfied.

Rubric:

- 2.7 If the student's solution is consistent with the published solution or the student's solution works using its reduction in Q25.2.
- 0 if the student's solution does not work at all.