ALG2 Assignment C

Answer the questions in the document!

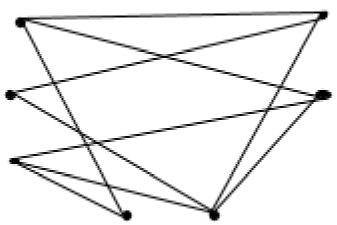
1. We consider the following Boolean formula given in CNF:

$$\left(x_1 \lor \overline{x_2} \lor x_3\right) \land \left(\overline{x_1} \lor x_2\right) \land \left(\overline{x_2} \lor x_3\right)$$

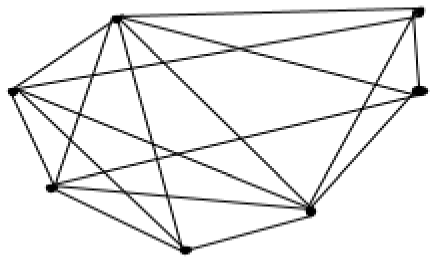
   There is a graph that corresponds to this formula in a way that each clique of size 3 of the graph defines a satisfying assignment for the given formula. Which of the following graphs is the corresponding graph:

   C

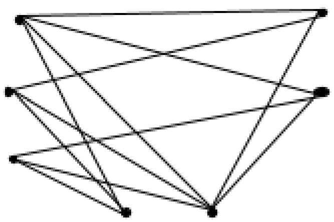
a.



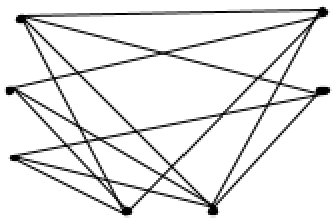
b.



c.



d.



2. Rewrite the following instances of SAT into 3SAT and check their satisfiability
   a. (A)
      (A∨A∨A) SAT
   b. $(A\lor \overline{B})\land(B\lor\overline{C} \lor \overline{D})$
      (A∨B∨x)∧(A∨B∨¬x) (B∨C∨¬D) SAT
   c. $(A\lor \overline{B})\land (A \lor C)\land (B \lor \overline{C})$
      (A∨B∨y)∧(A∨B∨¬y) (A∨C∨z)∧(A∨C∨¬z) (B∨¬C∨y) SAT
   d. $(A\lor B \lor C) \land (A\lor \overline{B}) \land (C\lor \overline{D} \lor \overline{G} \lor H)\land (B\lor C\lor D\lor G)$
      (A∨B∨C)∧(A∨B∨X)∧(A∨B∨¬X)∧(C∨D∨Y)∧(¬Y∨G∨H)∧(B∨C∨Z)∧(¬Z∨D∨G)

3. If we could find a polynomial time algorithm for an NP-complete problem would that mean:
   a. P = NP

    b. P ≠ NP
    c. none of the above

4. If we could find a polynomial time algorithm that solves an NP-complete problem in polynomial time on average would that mean:
    a. P = NP
    b. P ≠ NP
    c. none of the above

5. If we could show that the most efficient algorithm for finding clique in a graph requires exponential time would that mean:
    a. P = NP
    b. P ≠ NP
    c. none of the above

6. If we could show that vertex cover graph problem has an exponential number of potential solutions would that mean:
    a. P = NP
    b. P ≠ NP
    c. none of the above

7. Are the following statements true or false?
    a. Some NP-complete problems cannot be transformed into SAT problem in polynomial time (T/F) false
    b. A non-deterministic RAM may give different results for the same decision problem. (T/F) false
    c. A problem with exponential possible solutions can only be in P if P = NP. (T/F) false
    d. Every program that takes exponential time on a deterministic RAM can be made to run in polynomial time on a non-deterministic RAM. (T/F) false

8. A polynomial time algorithm that uses "if-better" function is simulated on a deterministic RAM. What is the complexity of the simulation with respect to the number "if-better" function is called in the algorithm?

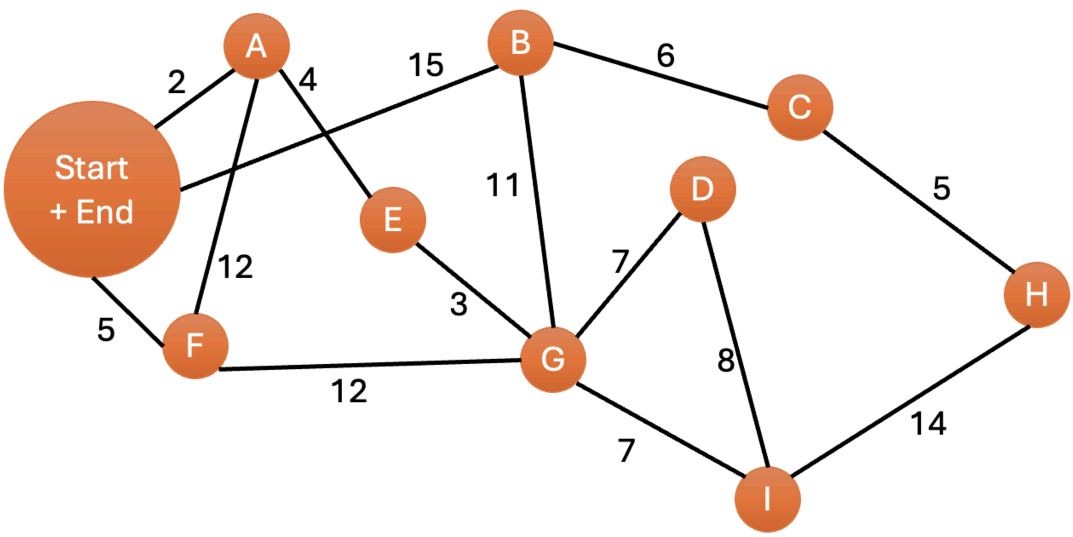
complexity simulation number of times "if-better" is called	Constant	Linear	Polynomial	Exponential
$O(1)$	x			
O(log(n))		x		
$O(n)$			x	

every 3rd instruction			x	

9. What would be sufficient evidence that could be checked in polynomial time that a Boolean formula is satisfiable
   a. A compete series of snapshots from a non-deterministic RAM solving SAT on the formula
   b. A list of satisfiable clauses
   c. A satisfying assignment for all the variables in the formula
   d. A satisfying assignment for 90% of the variables in the formula
   e. All calls to "if-better" on the Boolean formula

10. Are the following statements true or false?
    a. P=/=NP would mean that no NP-complete problem can be solved in polynomial time (T/F) True
    b. There are NP-complete problems that are "easier" to solve in practice than others (T/F) True
    c. The ability to solve some instances of an NP-complete problem implies P = NP (T/F) False
    d. Showing a problem is NP-complete means it cannot be solved in polynomial time (T/F) False

11. Consider the following weighted graph with vertices A, B, C, D, E, F, G, H, and I, and edges and weights as given below in the image. A weight denotes the time needed to drive from one house to another via the corresponding edge.



What is the most optimal tour when starting from the start and end point, driving along all 9 houses (a house can be visited more than once), and returning to the start and end point again, where optimal means the minimum driving time? Give the optimal tour and the optimal driving time. (Note: this is known as the shortest tour in a weighted graph problem! A tour is a walk through the graph without repeating edges.)

Start -> A (2)

A -> E (4)

E -> F (12)

F -> G (3)

G -> D (8)

D -> C (5)

C -> H (5)

H -> C (5)

C -> D (5)

D -> I (14)

I -> G (7)

G -> D (8)

D -> B (7)

B -> E (11)

E -> A (4)

A -> Start (2)

Total: 102

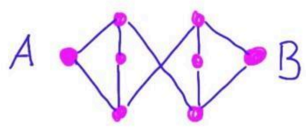
12. In a complete graph with n vertices, 1, 2, …, n, how many tours that visit all vertices exactly once[1] exists from vertex 1 to vertex n if:
    a. n = 5 6
    b. n = 10 40320

13. How the shortest tour problem can be stated as a decision problem:
    a. “Given a graphs what is the shortest tour?”
    b. “Given a graphs and a number d, is there a tour with a total length at most d”?
    c. “Given a graph what is the total length of the shortest tour?”

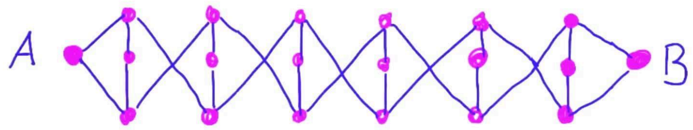
[1] A Hamiltonian path is a path that visits all vertices in a graph exactly once.

14. In a graph where vertices represent cities and edges represent direct connections and distance between pairs of cities, the Traveling salesmen problem is the problem of finding
    a. the shortest possible route that connects two given cities
    b. the shortest possible route that goes over all cities
    c. the shortest possible route that visits all cities exactly once
    d. the shortest possible route that visits all cities exactly once and returns to the original city.

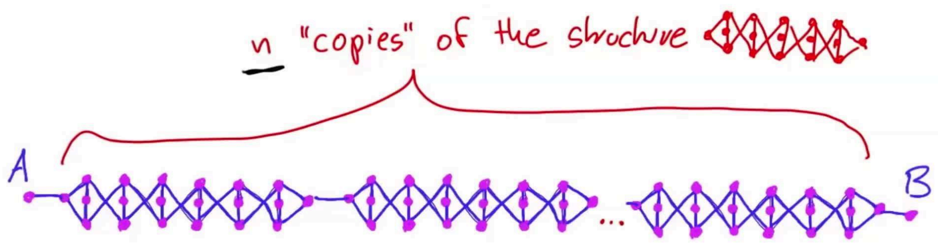
15. What is the length of the shortest route from A to B that visits all vertices? How many are they?
    a. 3 edges,2 routes



    b. 7 edges 2 routes



    c. 2n+1 edges, 2 routes



16. You encounter a problem that deals with
    a. Clustering/Finding groups of related objects clique
    b. "Covering" a set of objects vertex cover
    c. Optimizing a pathways shortest path
    d. Diversification/independent objects independent set

Which NP-complete problem would you choose as a base for reduction to show that your problem is NP-complete? You may choose from SAT, the shortest path that visits all vertices, clique, vertex cover, or independent set.

17.
   a. What is the longest common subsequence for strings *lemonade* and *blender*?
      lend
   b. Is the problem of finding the longest common subsequence of two strings NP-complete?
      no

18. Clique and k-colouring[2]:
   a. If a graph contains a clique of size k, then how many colours are needed to colour the graph? k colors
   b. If a graph can be coloured with at least k colours, what is the size of the largest clique in the graph? at most k
   c. Show a and b by drawing an example.

19. Transform the following instance of SAT into Clique and indicate a possible maximum Clique
   a. $(A\vee B\vee C)\wedge(A\vee \overline{B})\wedge(\overline{A}\vee \overline{B}\vee D)\wedge(C\vee \overline{D})$
      (A⋁B⋁C)⋀(A⋁¬B)⋀(A⋁B⋁D)⋀(C⋁D)

20. Transform the following instance of SAT into an Independent Set and indicate a possible maximum independent set
   a. $(A\vee B\vee C)\wedge(A\vee \overline{B})\wedge(\overline{A}\vee \overline{B}\vee D)\wedge(C\vee \overline{D})$
      (A∨B∨C)∧(A∨¬B)∧(A∨B∨D)∧(C∨¬D)
   b. Transform this into vertex cover.

[2] No edge connects vertices with the same colour.