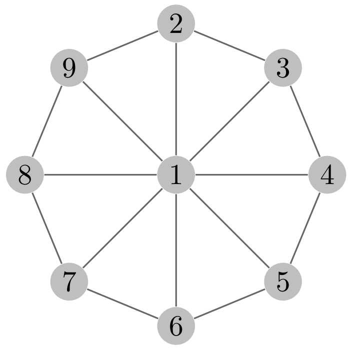
Semester Examination

Write your roll number in the space provided on the top of each page. *Write your solutions clearly in the space provided after each problem. You may use additional sheets for working out your solutions; attach such sheets at the end of the question paper.* **Attempt all problems.**

Name and Roll Number: ______________________________

Problem	Points	Score
1	6	
2	7	
3	9	
4	18	
5	20	
6	20	
7	20	
Total:	100	

1. Consider the following undirected wheel graph.



Assume that the above wheel graph is given as adjacency lists, where in the list of vertex v, the neighbours of v are listed in ascending order.

(a) Suppose a depth-first search (DFS) is performed on this graph with vertex 1 as root. Then, in the resulting DFS tree: $\boxed{3}$
the number of back-edges is _____________;

the number of tree-edges is _____________;

the number of leaves (not counting the root) is _____________.

(b) Suppose a breadth-first search (BFS) is performed on this graph, with vertex 2 as root. Draw the BFS tree. $\boxed{3}$

2. Suppose we are given two polynomials of degree at most d with integer coefficients:

$$A(X) = \sum_{i=0}^{d} a_i X^i \qquad\qquad B(X) = \sum_{i=0}^{d} b_i X^i.$$

Let $C(X) = A(X)B(X)$. Then, $C(X)$ is a polynomial of degree at most $2d$, say $C(X) = \sum_{i=0}^{2d} c_i X^i$. We wish to obtain the coefficients of $C(X)$ using the Fast Fourier Transform (FFT) algorithm. Let $M_n = (m_{ij} : i, j = 0, 1, \ldots, n-1)$ be the $n \times n$ FFT matrix, where n is a power of 2, ω is a primitive n-root of unity and $m_{ij} = \omega^{ij}$. Assume $n > d$, and let the column vectors $v_a, v_b \in \mathbb{R}^n$ be defined by

$$v_a = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_d \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \qquad v_b = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_d \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Fill in the blanks so that the steps below compute the coefficients of C correctly. Note that α, β and γ below are vectors in $\mathbb{C}^n$.

(a) Let $\alpha = M_n v_a$; *This step requires* $O($____________$)$ *multiplications using FFT.* $\boxed{1}$

(b) Let $\beta = M_n v_b$; *This step requires* $O($____________$)$ *multiplications using FFT.* $\boxed{1}$

(c) Let $\gamma = (\gamma_i : i = 0, 1, \ldots, n-1)$ be an n-dimensional column vector, $\boxed{1}$

where $\gamma_i =$ ____________, for $i = 0, 1, \ldots, n-1$. (Fill in the blank, keeping in mind the use of γ in the next step.)

(d) Then the coefficients corresponding to C, namely $c_0, c_1, \ldots, c_{2d}$, are obtained as the first $2d$ entries of the vector $M_n^{-1}\gamma$. $\boxed{2}$

This step requires $O($____________$)$ *multiplications.*
Explain in two sentences how you will use FFT for this step.

(e) *For this method to work correctly, we should pick* $n \geq$ ____________ *(write an expression in terms of d). Explain in one sentence.* $\boxed{2}$

3. Recall the representation of a min-heap with n elements as an array A (the indices run from 0 to $n-1$): the value at the root is $A[0]$, the two children of $A[i]$ are $A[2i+1]$ and $A[2i+2]$. We considered the following operations on heap: (i) bubbleup(i), which starts at $A[i]$ and repeatedly swaps the element with its parent whenever the parent's value is larger; (ii) bubbledown(i), which repeatedly starts at $A[i]$ and repeatedly swaps the element with the smaller of its two children, until the element reaches a location where it is at least the value of its children.

(a) Suppose the array A is turned into a min-heap by applying the operations bubbleup(i), for $i = 0, 2 \ldots, n-1$. How many comparisons will be performed in the worst case? $\boxed{2}$

Answer: θ(______________)

(We say that $f = \theta(g)$ if $f = O(g)$ and $g = O(f)$; so the answer you write down should be both an upper bound and a lower bound, ignoring constant factors.)

(b) Suppose the array A is turned into a heap by applying the operations bubbledown(i), for $i = n-1, \ldots, 0$. How many comparisons will be performed in the worst case? $\boxed{2}$

Answer: θ(______________)

(c) Consider the following array with ten characters. $\boxed{5}$

0	1	2	3	4	5	6	7	8	9
C	O	M	P	L	E	X	I	T	Y

Suppose this array is turned into a min-heap using the second method (using bubbledown). What heap will be produced

0	1	2	3	4	5	6	7	8	9

4. (a) Let $G = (V, E)$ be a directed graph with vertex set $\{1, 2, \ldots, n\}$. The transitive closure of G is the graph $G^* = (V, E^*)$ such that $\boxed{5}$

$$E^* = \{(u, v) \in V \times V : v \text{ is reachable from } u \text{ in } G\}.$$

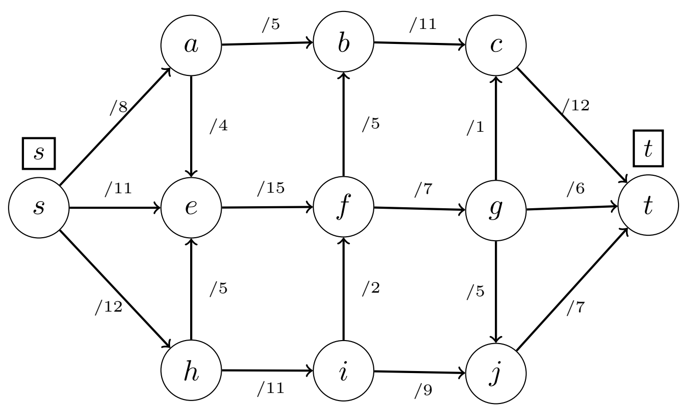
Fill in the blanks in the recurrence below such that $E_n = E^*$.

$$E_0 = E \cup \underline{\hspace{10em}};$$

and for $i = 1, 2, \ldots, n$,

$$E_i = E_{i-1} \cup \{(k, \ell) : (k, i) \in \underline{\hspace{3em}} \text{ and } (i, \ell) \in \underline{\hspace{3em}}\}.$$

(b) In the following *directed* network with capacities, determine a maximum (s, t)-flow and a minimum (s, t)-cut. Write the flow on the edges; write s or t next to each vertex to indicate on which side of the cut the vertex falls (the vertices s and t already have $\boxed{\text{s}}$ and $\boxed{\text{t}}$ written next to them). Note that a label of a/c on an edge indicates that the edge carries a flow of a and its capacity is c.) $\boxed{8}$



The value of the flow is __________. The capacity of the cut is __________.

(c) In class we observed that for every circuit $C(X)$ with inputs $X = (X_1, X_2, \ldots, X_n)$ we can efficiently obtain a 3-SAT expression (an AND of ORs, where each OR has AT MOST three literals) $\Phi(X, Y)$, where $Y = (Y_1, Y_2, \ldots, Y_m)$ are new variables, such that $\boxed{5}$

$$\forall x \in \{0,1\}^n \, [C(x) \text{ iff } \exists y \in \{0,1\}^m \, \Phi(x, y)] \, .$$

Suppose $C^*(X_1, X_2, X_3, X_4) = X_1 \vee X_2 \vee X_3 \vee X_4$. Write down an appropriate 3-SAT expression Φ^* corresponding to $C^*(X_1, X_2, X_3, X_4)$.

$\Phi^*(X_1, X_2, X_3, X_4, Y_1, Y_2, Y_3) =$ ______________________________________

______________________________________.

(It is OK if you can manage with fewer Y's.)

5. Consider the following SubsetSum problem.
Input: Positive integers $s_1, s_2, \ldots, s_n$, C.
Output: A subset $T \subseteq \{1, 2, \ldots, n\}$ such that $\sum_{i \in T} s_i = C$, if one exists; otherwise say *no solution exists.*

Describe a dynamic programming solution for the SubsetSum problem.

(a) Define a suitable predicate $A(j, c)$ for $j = 1, 2, \ldots, n$ and $c = 0, 1, \ldots, C$, and write a recurrence (including the base case) for it. [Also, write in words what $A(j, c)$ stands for.] $\boxed{5}$

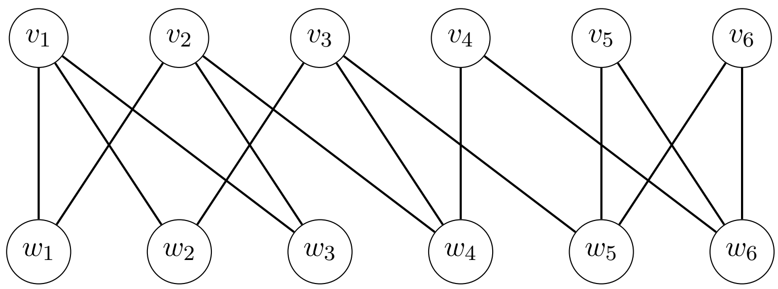
(b) Write a short program for computing $A(j, c)$ for various values of j and c. (You may use explicit `for` loops, or a recursive program with memoization.) $\boxed{5}$

(c) State how you would use/modify the computation in the previous part to obtain a solution T if it exists. $\boxed{4}$

(d) What is the running time of your solution? $\boxed{3}$

(e) Does this show that SubsetSum is in P? Explain. $\boxed{3}$

6. Consider the following bipartite graph.



(a) In the bipartite graph, show (by shading the appropriate edges and vertices), a maximum matching and a minimum vertex cover. (Work out the problem in a rough sheet and then copy the answer on the picture above, or attach a separate page with your solution.) $\boxed{6}$

(b) Is the maximum cardinality matching in this graph unique? $\boxed{2}$

(c) For a bipartite graph $H = (V, W, E)$, an edge $e \in E$ is said to be a *essential* if it is in every maximum cardinality matching in H. Determine the essential edges for the graph shown above. Does every bipartite graph have an essential edge? $\boxed{3}$

(d) Suppose a bipartite graph $H = (V, W, E)$ is given as input in the form of adjacency lists. In addition, suppose a perfect matching M in H is given. Describe an efficient algorithm (state its running time) to determine all the essential edges of H. (Hint: After removing an edge of the matching from the graph, how will you determine if the current matching can be improved?) $\boxed{9}$

7. Recall that a clique in an undirected graph $G = (V, E)$ is a subset $C \subseteq V$, such that

$$\{\{v, w\} : v, w \in C, v \neq w\} \subseteq E.$$

Consider the following CLIQUE problem on undirected graphs.
Input: A graph G on n vertices and a positive integer $k \leq n$.
Output: 1 iff G has a clique C with at least k elements.

(a) Argue that CLIQUE is in NP. $\boxed{1}$

(b) Show a reduction from 3-SAT to CLIQUE. That is, given a Boolean expression $\boxed{5}$

$$\Phi(X_1, X_2, \ldots, X_3) = (L_{11} \vee L_{12} \vee L_{13}) \wedge (L_{21} \vee L_{22} \vee L_{23}) \wedge \cdots \wedge (L_{n1} \vee L_{n2} \vee L_{n3}),$$

state how you will construct a pair (H, k) such that H has a clique of size k iff Φ is satisfiable.

(c) Show a direct reduction (without appealing to the Cook-Levin theorem) from CLIQUE to SAT by completing the following. $\boxed{6}$

Assume the input pair is (G, k), where $V(G) = \{1, 2, \ldots, n\}$ and $k \leq n$ (otherwise output $X_1 \wedge \overline{X}$, which is clearly unsatisfiable). Let $X_1, X_2, \ldots, X_n$ be the variables, where X_i is intended to indicate if vertex i is in the clique. The expression Φ is defined as follows.

$$\phi_{ij} = \underline{\hspace{6cm}} \quad \text{if } 1 \leq i < j \leq n \text{ and } \{i, j\} \notin E; \tag{1}$$

$$\Phi = \left(\bigwedge_{1 \leq i < j \leq n : \{i,j\} \notin E} \phi_{ij} \right) \wedge \Psi(X_1, X_2, \ldots, X_n, Y_1, \ldots, Y_m), \tag{2}$$

where $\Psi(X_1, X_2, \ldots, X_n, Y_1, \ldots, Y_m)$ is an expression with the following property: for all $x_1, x_2, \ldots, x_n\{0, 1\}^n$, there is a choice $y_1, y_2, \ldots, y_m \in \{0, 1\}^m$ satisfying Ψ iff $\sum_i x_i \geq k$. Assuming that the expression Ψ can be constructed efficiently (see part (d) below), show that Φ is satisfiable iff G has a clique of size k.

(d) Show how to construct $\Psi(X_1, X_2, \ldots, X_n, Y_1, \ldots, Y_m)$. [Hint: Consider the complete bipartite graph $H = (U, W, F)$ with $U = \{u_1, u_2, \ldots, u_n\}$ (one for each of the n variables $X_1, X_2, \ldots, X_n$)), $W = \{w_1, w_2, \ldots, w_k\}$ and $F = U \times W$. For each edge e define a variable Y_e and write down a constraints so that $(Y_e : e \in F)$ corresponds to a matching of size k iff $\sum_{i=1}^{n} X_i \geq k$. Note that apart from the constraints mentioned in the quiz discussed in class, you might have to add constraints that link variables X_i and Y_{ij}.] $\boxed{8}$