1. Approximation algorithms:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>9</td>
<td>12</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>9</td>
<td>10</td>
<td>8</td>
<td>11</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

(a) Draw the List-Scheduling-schedule as a Gantt-Chart. How much is $C_{max}$ and how much is $\sum C_j$?

(b) Draw the LPT-schedule as a Gantt-Chart. How much is $C_{max}$

(c) Draw the SPT-schedule as a Gantt-Chart. How much is $C_{max}$ and how much is $\sum C_j$?

Solution:

(a) The Gantt-Chart for List-Scheduling is:

(b) The Gantt-Chart for LPT is:

Now, our $C_{max} = \max\{12, 29\} = 29$ and $\sum C_j = 78 + 56 + 62 = 196$.

(c) The Gantt-Chart for SPT is:

Now, our $C_{max} = \max\{12, 29\} = 29$ and our $\sum C_j = 85 + 85 + 82 = 252$. 
(c) The Gantt-Chart for SPT is:

Our $C_{\text{max}} = \max\{12, 31\} = 31$ and $\sum C_j = 45 + 59 + 64 = 168$.

(d) The lower bound on the makespan (see next question) is $84/3 = 28$. We can, in fact, achieve this by scheduling $C, J, F$ on $P_1$, $H, B, G$ on $P_2$ and the rest on $P_3$.

The worst has makespan 36. If we don’t consider the scheduling of the largest ($C$) job, then the remaining processing time has a makespan with lower bound of $72/3 = 24$. Suppose we schedule these optimally, then place $C$ on the machine with this makespan, we get a makespan of 36.

(e) SPT minimizes the sum of completion times at 168 and LPT has the maximum at 252.

2. Approximation algorithms

(a) Explain why $\sum_{j=1}^{n} p_j/m$ and $p_{\text{max}} = \max\{p_j\}$ are Lower Bounds for the makespan of any algorithm solving $P||C_{\text{max}}$.

Solution: Since $\sum_{j=1}^{n} p_j$ is the total amount of processing that must be done across all machines, $\sum_{j=1}^{n} p_j/m$ constitutes the average processing to be done over all machines. By properties of the average, at least one machine then must process at least this average. Therefore, for some machine, some $C_i \geq \sum_{j=1}^{n} p_j/m$.

Since a single job can only be processed by at most a single processor at a time, the job can be completed no faster than its given processing time.

(b) Find Upper Bounds for the makespan of List Scheduling for $P||C_{\text{max}}$. Solution:

One upper bound is the sum of all processing times (e.g. if they ran sequentially on a single machine). We can go further. Let $J_k$ denote the job that completes last across all machines, say on machine $P_t$. This job (by definition) was assigned when machine $P_t$ had the least load, so that:

$$C_{\text{max}} = C_t = p_k + s_k$$

where $s_k$ is the amount of load on machine $P_t$ when $J_k$ was assigned to it. But this, by definition, must have been the least load among the machines when this happened. Therefore (regardless of assignment order) by part (a):

$$s_k \leq \sum_{j \neq k} p_j/m \leq \left(1 - \frac{1}{m}\right) C_{\text{max}}$$

and also by part (a):

$$p_k \leq C_{\text{max}}$$

to combine to find:

$$C_{\text{max}} \leq \left(2 - \frac{1}{m}\right) C_{\text{max}}$$
3. **Graphs and their representation, DFS, BFS.** The third question refers to the following directed graph:

Answer the following questions:

(a) Give the degrees of the vertices $A$, $B$, and $E$.
   **Solution:** $\deg(A) = 4$, $\deg(B) = 6$, $\deg(E) = 5$ (loops are double-counted; definition 3, page 652)

(b) What are the direct neighbors of vertex $B$?
   **Solution:** $A$, $C$, $D$, $E$

(c) Are there any loops in the graph?
   **Solution:** Yes. ($C$ and $E$)

(d) Are there any cycles in the graph? What about simple cycles?
   **Solution:** Yes and yes. Simple cycle: $A \to B \to D \to A$. Every simple cycle is a circle. An example of a cycle that isn’t simple would be $A \to B \to D \to A \to B \to D \ldots$ there are a lot of examples of both in this graph.

(e) Find a trail from vertex $A$ to vertex $E$.
   **Solution:** $A \to B \to E$

(f) Find a path from vertex $A$ to vertex $E$ that is no trail.
   **Solution:** $A \to B \to D \to A \to B \to E$

(g) Is this graph a simple directed graph? Explain why or why not.
   **Solution:** No. Rosen pg 643: simple directed graph has no loops/no multiple directed edges.

(h) Does this graph have a source and/or a sink? Explain why or why not.
   **Solution:** No to both. Sinks have no outgoing edges and sources have no incoming edges. There are no vertices that satisfy either of those definitions in this graph.

(i) Is this graph (strongly or weakly) connected or disconnected? Explain your answer.
   **Solution:** Strongly connected. Rosen pg 685: a strongly connected graph has a path from $u$ to $v$ and $v$ to $u$ for any vertices $u$ and $v$ in the graph. The entire graph is one giant cycle so clearly you can get from any vertex to another and back.

(j) Give the adjacency matrix representation of the graph.
   **Solution:**
   
   \[
   \begin{bmatrix}
   0 & 1 & 0 & 1 & 0 \\
   1 & 0 & 1 & 1 & 1 \\
   0 & 1 & 1 & 0 & 0 \\
   1 & 0 & 0 & 0 & 1 \\
   0 & 0 & 1 & 0 & 1 
   \end{bmatrix}
   \]

(k) Give the adjacency list representation of the graph.
   **Solution:**
   
   \[
   \begin{align*}
   A : & \quad B, D \\
   B : & \quad A, C, D, E \\
   C : & \quad B, C
   \end{align*}
   \]
D : A, E
E : C, E

(l) Let A be the starting node. Traverse the graph with Depth-First-Search (DFS) and give each node a number when it is detected (for example, A gets number 1, the next node that is detected gets number 2 and so on). Always break ties in alphabetical order (for example, if you must choose between B and C, choose B).

**Solution:** A − 1, B − 2, C − 3, E − 4, D − 5 Graphical solution at the end.

(m) Do the same as in the question before, but now with Breadth-First-Search (BFS).

**Solution:** A − 1, B − 2, D − 3, C − 4, E − 5 Graphical solution at the end

4. **Graphs and their representation.** The second question refers to the following undirected graph:

![Undirected Graph](image)

Answer the following questions:

(a) Give the degrees of the vertices A, B, and E.

**Solution:** \( \text{deg}(A) = 2, \text{deg}(B) = 2, \text{deg}(E) = 2 \)

(b) What are the direct neighbors of vertex B?

**Solution:** A and B

(c) Are there any loops in the graph?

**Solution:** No.

(d) Are there any cycles in the graph? What about simple cycles? | **Solution:** Yes and yes.

(e) Find a trail from vertex A to vertex E.

**Solution:** A → C → F → E

(f) Find a path from vertex A to vertex E that is no trail.

**Solution:** A → B → C → A → B → C → F → E

(g) Is this graph a simple undirected graph? Explain why or why not.

**Solution:** Yes.

(h) Does this graph have a source and/or a sink? Explain why or why not.

**Solution:** No and no. Impossible for an undirected connected graph to have either a source or as sink.

(i) Is this graph (strongly or weakly) connected or disconnected? Explain your answer.

**Solution:** Strongly connected. It’s impossible for an undirected connected graph to not be strongly connected.

(j) Give the adjacency matrix representation of the graph.

**Solution:**

\[
\begin{bmatrix}
0 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0
\end{bmatrix}
\]
(k) Give the adjacency list representation of the graph.

**Solution:**
A : B, C  
B : A, C  
C : A, B, F  
D : E, F  
E : D, F

5. **Some graph properties:** Prove the following claims.

(a) Let \( G = (V, E) \) be an undirected graph with \( |E| \) edges. Then \( \sum_{v \in V} deg(v) = 2|E| \).

**Solution:** Note that every edge counts towards degrees twice: once for the vertex it leaves and one for the vertex it enters. Therefore, the sum of degrees is exactly the twice the sum of edges.

(b) An undirected graph has an even number of vertices of odd degree.

**Solution:** The previous part tells us that the sum of the degrees is even. Then, the sum of the degrees for the graph can be broken into the sum of vertices with odd degrees and the sum of even degrees. However, a sum of even numbers is always even, so that the sum of the odd degrees must also be even. This only happens if there is an even number of vertices with odd degree.

(c) Let \( G = (V, E) \) be a directed graph with \( |E| \) edges. Then \( \sum_{v \in V} deg^-(v) = \sum_{v \in V} deg^+(v) = |E| \).

**Solution:** Every directed edge has exactly one vertex from which it leaves and exactly one vertex to which it enters. The sum of the in-degrees across all vertices must be exactly the sum of out-degrees across all vertices.

6. **Trees.** Draw a full binary tree of height 3. Number the nodes alphabetically starting with A in the root and then level for level from the left to the right in increasing order.

**Solution:**

![Tree Diagram]

(a) What is the parent of node L?

**Solution:** F

(b) What are the children of node C?

**Solution:** F and G

(c) Is the tree balanced? Explain your answer.

**Solution:** Yes. Every leaf node is at depth of \( h \) or \( h - 1 \).

(d) Name the leaves of tree. How many leaves does this tree have? Can you generalize your answer?

**Solution:** Leaves: H, I, J, K, L, M, N, O. This tree has 8 leaves. Number of leaves in a full binary tree is \( 2^h \).
(e) How many internal vertices does the tree have? Can you generalize your answer?

**Solution:** This tree has 7 internal nodes. Number of internal nodes in a full binary tree is $2^h - 1$. 
7. For each statement, either prove that it is true or give a counterexample to show that it is false.

(a) Every tree on $n$ nodes has exactly $n - 1$ edges.

**Solution:** We can prove this using induction on $n$, the number of nodes in the graph. For the base case, when $n = 1$, a tree with one node has no edges.

For the induction hypothesis, assume that any tree of $k$ nodes has $k - 1$ edges. We will prove that any tree $T$ of $k + 1$ nodes has $k$ edges. Let $u$ be a degree one node, or a leaf of the tree. We proved in class that all trees have at least one node which has degree one. Consider the graph $T' = T - u$, which is obtained by removing $u$ and its only incident edge from $T$. $T'$ is still a tree and it has $k$ nodes. So, it must have $k - 1$ edges from induction hypothesis. Now, if we add $u$ back, we are adding one more edge. So, number of edges in $T$ is $k$. This proves the statement by induction.

(b) Every graph with exactly $n - 1$ edges is a tree.

**Solution:** This statement is false. A counter-example would be for $n = 4$, a triangle with an isolated vertex. That is, $V = v_1, v_2, v_3, v_4$ with the set of edges, $E = (v_1, v_2), (v_3, v_2), (v_1, v_3)$. Here, $n = 4$, there are $n - 1 = 3$ edges, but the graph is not a tree because it has a cycle.

However, the statement is correct if the graph is connected. That is, any connected graph of $n$ vertices with $n - 1$ edges is always a tree. See if you can try to prove it [Hint: Prove that there should be a leaf node and use induction].

8. Euler and Hamiltonian Tours.

(a) Describe briefly what the problem of the 7 Bridges of Königsberg is.

**Solution:** The 7 Bridges of Königsberg problem asks whether or not it is possible to start at a certain place in the town of Königsberg, cross all 7 bridges exactly once each, and return to the same place in town.

(b) What is an Euler tour? What is an Euler Cycle? What is a Hamiltonian Tour? What is a Hamiltonian Cycle? Is it (from a computationally point of view) harder to compute Hamiltonian Tours than to compute Euler Tours? Explain why or why not.

**Solution:** An Euler Cycle is a traversal of the graph starting at some vertex $v$, traversing every edge exactly once, and ending at $v$. An Euler Tour relaxes the condition of ending at $v$.

A Hamiltonian Cycle is likewise a traversal of a graph starting at some vertex $v$, but instead requires hitting every single vertex exactly once before returning back to $v$. Again, a Hamiltonian Tour relaxes the condition of returning to $v$.

Computationally, Hamiltonian Tours are significantly more difficult than Euler Cycles. Euler Cycles can be done in polynomial time where as Hamiltonian Tours take exponential time.

(c) True or false? If a graph $G$ has 3 or more even vertices $\leftrightarrow$ $G$ has no Euler tour.

**Solution:** False. Euler Tour requires that at most 2 vertexes can have odd degrees. A counter example would thus simply be a graph with 3 vertexes with even degrees and 3 vertexes with odd degrees.

9. Modelling problems with graphs: Hamiltonian/Euler. A ballet recital consists of some number of different acts, each containing several dancers. Suppose you are given a list of all the acts in the recital, with the names of the dancers that will appear in each act. Some dancers may be in more than one act, but each act requires a different costume. To allow the dancers time to change costumes, the recital should be set up so that no dancer is in two consecutive acts. Our goal is to find an ordering of the acts for the recital so that no dancer is in back-to-back acts.

(a) Describe how to model this situation using a graph. Carefully describe what the vertices of your graph represent, and when two vertices are connected with an edge.

**Solution:** The vertices are the acts of the show. Connect two vertices (acts) with an undirected edge if they do not share any dancers.
(b) Now that you’ve modeled the situation with a graph, which problem from graph theory are you trying to solve on this graph?

**Solution:** We want to find a path that uses every vertex exactly once, or a Hamiltonian tour.

(c) Is it always possible to achieve the goal? Explain.

**Solution:** No. If the show has only two acts and there is somebody in both acts, we cannot achieve the goal. In other words, not every graph has a Hamilton tour.

10. **Modelling problems with graphs: Hamiltonian/Euler.** We say a matrix has dimensions $m \times n$ if it has $m$ rows and $n$ columns. If matrix $A$ has dimensions $x \times y$ and matrix $B$ has dimensions $z \times w$, then the product $AB$ exists if and only if $y = z$. In the case where the product exists, $AB$ will have dimensions $x \times w$. In this problem, we are given a list of matrices and their dimensions, and we want to determine if there is an order in which we can multiply all the matrices together, using each matrix exactly once. For example, here is a possible list of matrices and their dimensions:

$$
\begin{align*}
\text{A} & \text{ is } 3 \times 2 \\
\text{B} & \text{ is } 6 \times 3 \\
\text{C} & \text{ is } 2 \times 5 \\
\text{D} & \text{ is } 5 \times 3 \\
\text{E} & \text{ is } 3 \times 6 \\
\text{F} & \text{ is } 6 \times 2
\end{align*}
$$

(a) Given any list of matrices and dimensions, describe how to draw a graph so that each order in which we can multiply the matrices corresponds to a Hamiltonian tour of your graph. Carefully describe what the vertices of your graph represent, and when two vertices are connected with an edge.

**Solution:** Let each matrix in the list be a vertex. Draw a directed edge from matrix $M$ to matrix $N$ if the number of columns of matrix $M$ equals the number of rows of matrix $N$, or in other words, if the matrix product $MN$ is defined.
(b) Draw the graph described in part (a) for the example list of matrices given above.

Solution:

(c) Given any list of matrices and dimensions, describe how to draw a graph so that each order in which we can multiply the matrices corresponds to an Eulerian tour of your graph. Carefully describe what the vertices of your graph represent, and when two vertices are connected with an edge.

Solution: Make a vertex for each number that appears as a dimension of a matrix in the list. Draw a directed edge from vertex $i$ to vertex $j$ if there is an $i \times j$ matrix in the list.

(d) Draw the graph described in part (c) for the example list of matrices given above.

Solution: The labels are optional, but helpful in solving the problem.

(e) For the given example list of matrices, give one order in which we can multiply those matrices, or say that no such order exists.

Solution: There are three possible answers:

- $FCDEBA$
- $BEFCDA$
- $BACDEF$

11. **Modeling problems with graphs.** Say that two actors are co-stars if they have been in the same movie. Show that in any group of six actors, we can either find a group of three such that all pairs in the group are co-stars, or a group of three so that no two in the group are co-stars.

Solution: We want to show that in any group of six actors, at least one of these situations occurs:

(a) There is a group of three actors such that all pairs in the group are co-stars.

(b) There is a group of three actors so that no two in the group are co-stars.

Choose one particular actor, who we will call actor A. Exactly one of these two things is true:

(i) Actor A has 3 or more co-stars among the group of 6.

(ii) Actor A has less than 3 co-stars among the group of 6.

In Case (i), consider the co-stars of actor A. If no two of them have been co-stars of one another, then since there are at least three of them, this forms a group of three such that no two in the group are co-stars, situation (b). If some two of them have been co-stars of one another, then those two together with actor A form a group of three such that all pairs in the group are co-stars, situation (a).

In Case (ii), if A has less than 3 costars among the group of 6, that means there are at least three people with whom he is not a co-star. Consider these non-co-stars of actor A. If all of them have been
co-stars of one another, then since there are at least three of them, this forms a group of three such that all pairs in the group are co-stars, situation (a). If some two of them are not co-stars, then those two together with actor A form a group of three in which no two in the group are co-stars, situation (b).

Thus, in all cases, we can always find situation (a) or (b), which is what we were trying to prove.

Note: We can formulate our answer as a problem of graph theory by constructing an undirected graph as follows. Let the vertex set V be the set of six actors. Connect two actors with an edge if they are co-stars. We must show that one of the following two situations occurs:
(a) The graph contains a triangle, i.e. three vertices which are all connected.
(b) The graph contains an anti-triangle, i.e. three vertices, none of which are connected.

Choose one particular vertex, let’s say v, and then the cases become:
(i) degree(v) ≥ 3
(ii) degree(v) < 3

The argument is exactly the same, but viewing it as a problem about graphs might help you to visualize what is going on, and it also frames the question in terms of abstract objects and relationships, not just the special case of actors with a co-star relationship that we have addressed here.