All problems have the same weight. Write the solutions in separate sheets for each problem.

Problem 1 (Enumeration)**.** Let $n \geq m$ be positive integers. We have m tasks $t_1, \dots, t_m$ and each one is assigned to one of n processors $p_1, \dots, p_n$. How many assignments are there if:

(a) Each processor can receive at most one task?

(b) Each processor can receive at most one task but the tasks are indistinguishable?

(c) Each processor can receive any number of tasks?

In each case, give the answer **both** in general and for the special case $m = 3$, $n = 5$.

Solution. (a) Task t_1 can be assigned to n processors. For each choice t_2 can be assigned to one of the $n-1$ remaining processors and so on: it is the number of ordered samples of m out of n processors without replacement. The number is

$$n(n-1)\cdots(n-m+1).$$

For $n = 5$ and $m = 3$ the number is $5 \cdot 4 \cdot 3 = 60$.

(b) If we are only interested in whether a processor has an assigned task, we count the number of subsets of m processors out of n, or the number of unordered samples of m out of n processors without replacement. The number is

$$\binom{n}{m}$$

For $n = 5$ and $m = 3$ the number is $\binom{5}{3} = 10$.

(c) Every task can be assigned to any of the n processors, the number is now n^m, the number of ordered samples of m out of n processors with replacement. For $m = 3$ and $n = 5$ the answer is $5^3 = 125$.

Problem 2 (Recurrences)**.** Consider the growth of a population of rabbits consisting of young, mature, and old animals. Initially there are 16 young rabbits. At the end of each year, young animals reach maturity, while mature ones produce offspring of 9 young rabbits per mature one. Additionally, immediately after having offspring, mature rabbits grow old. Old animals do not produce any offspring.

(a) Formulate the recurrence relation for the total population at the **end** of each year. Give an explanation for your answer.

(b) Solve this recurrence using the tools from the lecture. *Hint:* You can use the fact that $x^3 - x^2 - 9x + 9 = (x-1)(x-3)(x+3)$.

Solution. (a) It is clear that if a_n denotes the total population at the end of year n, then

$$a_n = Y_n + M_n + O_n,$$

where Y_n, M_n, O_n are the number of young, mature, and old rabbits at the end of year n. Now notice that every animal that is old by the end of year n was already alive at the end of year $n-2$, and conversely, every animal that was alive at the end of year $n-2$ has grown old by the end of year n, that is, $O_n = a_{n-2}$. We can make a similar argument for the mature animals. Note that every rabbit that was alive in the previous year will now be either mature or old, that is $M_n + O_n = a_{n-1}$. Inserting our formula for O_n gives

$$M_n = a_{n-1} - O_n = a_{n-1} - a_{n-2}.$$

Finally, the number of young rabbits is exactly 9 times the number of mature ones in the previous year, and hence by our previous argument

$$Y_n = 9M_{n-1} = 9a_{n-2} - 9a_{n-3}.$$

Putting everything together, we see that

$$a_n = a_{n-1} + 9a_{n-2} - 9a_{n-3}.$$

What is left is to compute the initial values. Initially, that is at the end of year zero, we had 16 young animals. These grow old by the end of year one and become 16 mature rabbits. By the end of year two all 16 of them produce 9 offspring each and become old, and hence

$$a_0 = 16,\ a_1 = 16,\ a_2 = 160.$$

(b) The characteristic polynomial is $x^3 - x^2 - 9x + 9 = (x-1)(x-3)(x+3)$ so it has roots $\alpha = 1$, $\beta = 3$, and $\gamma = -3$. The general form of the recurrence will therefore be

$$a_n = A + 3^n B + (-3)^n C,$$

for some appropriate constants A, B, C that we can determine using our initial values. If we solve the linear system

$$\begin{aligned} 16 &= A + B + C \\ 16 &= A + 3B - 3C \\ 160 &= A + 9B + 9C \end{aligned}$$

we see that $A = -2$, $B = 12$, and $C = 6$, and so the general solution will be

$$a_n = -2 + 12 \cdot 3^n + 6 \cdot (-3)^n.$$

Problem 3 (Graphs)**.** Consider the graph H_n whose vertex set consists of all words of length n from the three letter alphabet $\{0, 1, 2\}$. Two words are adjacent if they differ in precisely one entry.

(a) Is H_n regular? If yes, what is the degree of its vertices? Give in particular the answer for $n = 4$.

(b) How many edges does H_n have? Give in particular the answer for $n = 4$.

(c) Is H_n connected? If yes, what is its diameter?

(d) Is H_n bipartite?

Solution. (a) The neighbors of a given vertex are the ones obtained by choosing a coordinate, n choices, and replacing it by a different symbol, two choices. The graph is regular of degree $d = 2n$.

(b) By the Handshaking Lemma $dN = 2m$, where $N = 3^n$ is the number of vertices and $m = dN/2 = n3^n$ is the number of edges.

(c) There is a path from a vertex $(x_1, \dots, x_n)$ to any other vertex $(y_1, \dots, y_n)$ by moving successively through the vertices

$$(x_1, \dots, x_n), (y_1, x_2, \dots, x_n), (y_1, y_2, x_3, \dots, x_n), \dots, (y_1, \dots, y_{n-1}, x_n), (y_1, \dots, y_n).$$

Therefore the graph is connected and, in particular, its diameter is at most n. Since the distance from $(0, 0, \dots, 0)$ to $(1, 1, \dots, 1)$ is precisely n, the diameter is n.

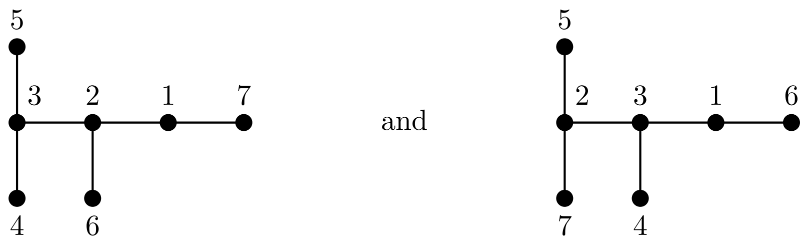
(d) The graph contains the cycle $(0, 0, \dots, 0), (1, 0, \dots, 0), (2, 0, \dots, 0)$ of odd length 3, so it is not bipartite.

Problem 4 (Trees)**.** Let T and T' be the trees whose Prüfer codes are $(3, 3, 2, 2, 1)$ and $(3, 2, 1, 3, 2)$, respectively.

(a) Draw the two trees T and T'. Are they isomorphic?

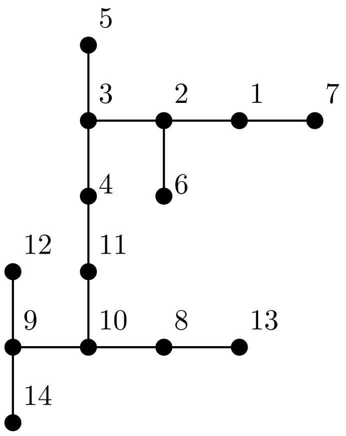
(b) Consider the graph G obtained from T and T' in the following way: First, relabel the vertices of T' by adding 7 to every current label. Then, draw an edge between vertex 4 of T and vertex 11 of T'. Is G a tree? If yes, compute its Prüfer code.

Solution. (a) The trees T and T' are



We see that the permutation $(1)(23)(467)(5)$ applied to the vertices of T results in the tree T', so they are isomorphic.

(b) The resulting graph is connected, has 14 vertices and 13 edges, so it is a tree. It looks like this:



Its Prüfer code is $(3, 2, 1, 2, 3, 4, 11, 10, 9, 8, 10, 9)$.

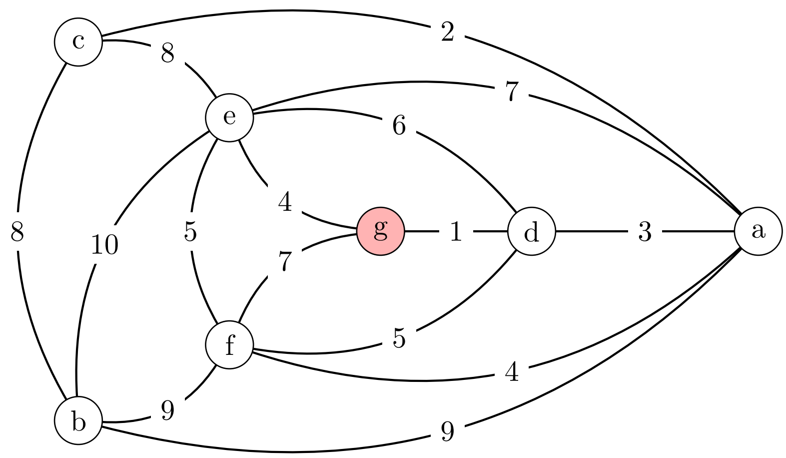
Problem 5 (Prim's algorithm)**.**

(a) What does Prim's algorithm return when given a connected graph $G = (V, E)$ with positive edge weights $w : E \to \mathbb{R}^+$ and a root vertex $x \in V$ as input? Make sure to define all terms you are using that were not mentioned here.

(b) Write down Prim's algorithm as `pseudocode`.

(c) Suppose Prim's algorithm is given a weighted connected graph $G = (V, E)$ and a root vertex $r \in V$ such that all the edge weights are positive and pairwise distinct. Prove that the output is unique. *Hint:* For this you do **not** need to show that the minimum spanning tree is unique.

(d) Write down (or draw) the output of Prim's algorithm applied to the following graph, starting from vertex g. You do not need to write down individual steps.



Solution. (a) Prim's algorithm returns the *minimum spanning tree* T of a given weighted connected graph $G = (V, E)$. A tree T is a connected graph without cycles. Spanning means that T has vertex set V, and minimum means that T has minimal edge weight among all spanning trees of G.

(b)

Set $V(T) = \{x\}$, $E(T) = \emptyset$.
while T does not contain every vertex of G **do**
    Let yz be the weight minimal edge in $\{ab \in E(G) : a \in V(T) \text{ and } b \in V(G) \setminus V(T)\}$.
    Update $V(T) = V(T) \cup \{y, z\}$ and $E(T) = E(T) \cup \{yz\}$.
end while
return T

(c) If the starting point is fixed and the edge weights are all unique, then in every iteration of the while loop there will be a unique minimum choice for the edge to add, and hence the output is unique.

(d) We get the following minimum spanning tree: