Solution set of exercises 01

Exercise 01



We represent the wagons by the vertices. An edge connects

two vertices i and j if wagons i and j cannot be on the same track.

We obtain the graph opposite. We see that 1, 3 and 5 cannot

be on the same track.

Therefore, at least three tracks are required.

Exercise 02

We obtain the following bipartite graph (left):





By coloring the edges of this graph (1 color = 1 hour of the schedule), taking care that each vertex does not have two incident edges of the same color, we obtain the result on the right. From this colored graph, we draw the following schedule:

|  |  |  |  |
| --- | --- | --- | --- |
|  | P1 | P2 | P3 |
| 1st hour (red) | C1 | C3 | C2 |
| 2nd hour (green) | C1 | C2 | C3 |
| 3rd hour (blue) | C2 | C1 | C3 |
| 4th hour (black) |  |  | C1 |

Exercise 03

We obtain the complete graph G6.



It will take 5 days of tournament. Here is a possible schedule:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Day 1 | Day 2 | Day 3 | Day 4 | Day 5 |
| 1-2 | 2-3 | 1-3 | 2-4 | 1-4 |
| 3-4 | 4-5 | 4-6 | 1-5 | 2-6 |
| 5-6 | 1-6 | 2-5 | 3-6 | 3-5 |

This schedule was constructed based on the five diagrams below:

Exercise 04

The movements are therefore (for example): c3-b1, a3-c2, a1-b3, c1-a2, b1-a3, c2-a1, b3-c1,

a2-c3, c3-b1, a3-c2, a1-b3, c1-a2, b1-a3, c2-a1, b3-c1, a2-c3.



Exercise 05

* Let G = (V, E) be a simple graph. When calculating the sum of the degrees of the vertices, each edge (x, y) of E is counted twice, once for d(x) and again for d(y). Therefore, this sum is finally equal to twice the number of edges.
* Let P be the set of vertices of even degree and I the set of vertices of odd degree of a simple graph G = (V,E). P and I form a partition of V. By the handshake lemma, we have:







Now 2 · |E| and are even integers. is also even, since it is the difference of two even integers. Now, each term of the sum  is odd. It can therefore only be even if the number of terms is even. We have thus shown that the cardinal of I is an even integer.

* If everyone has at least one friend in the assembly, this means that all the degrees of the vertices are between 1 and n−1. Since there are n vertices, by the principle of drawers, it is certain that at least two have the same degree, so that two people have the same number of friends. If a person has no friends, the degree of the corresponding vertex is 0. The degrees of the n−1 other vertices are between 1 and n-2. Same conclusion as in the first case. If several people have no friends, then they have the same number of friends, in this case 0!

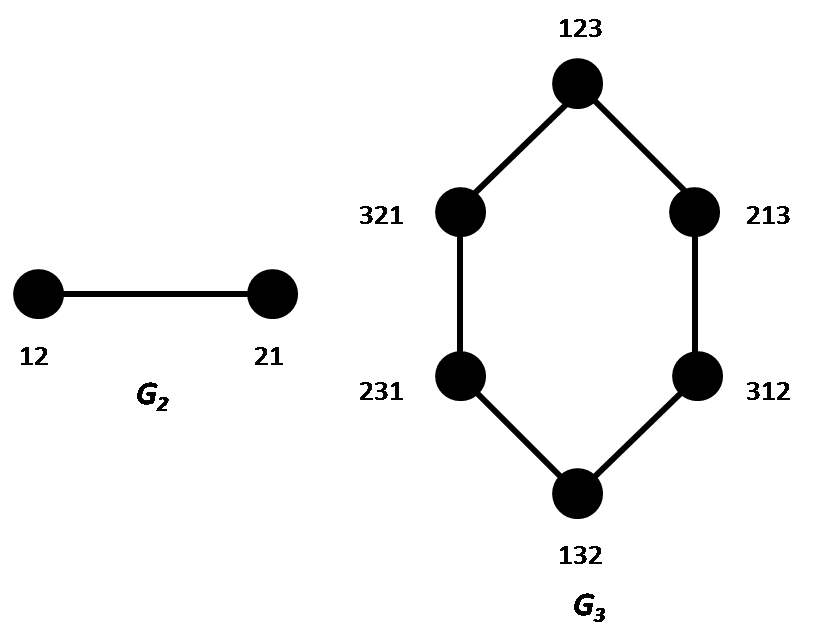
**Exercise 06**

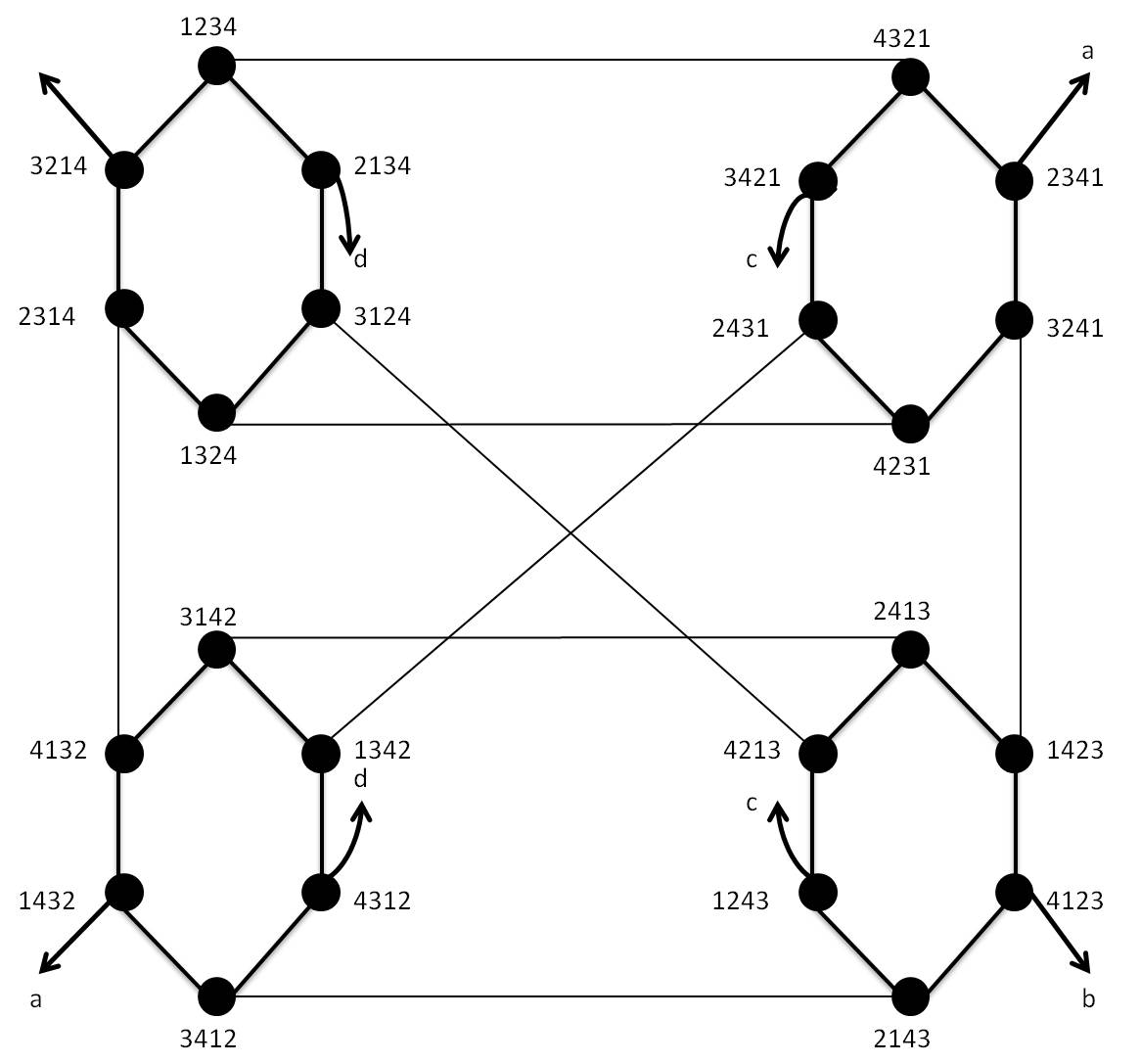
The n-Pancake graph:

Denoted G*n*, is formed from a set of n! nodes denoted V(G*n*) = {(u1, u2, ..., ui, ..., un) such that ui ∈ {1, 2, ..., n} and ui ≠ uj for i ≠ j}. The neighborhood of a node is defined as follows:

* A node (u1, u2, ..., ui, ..., un) is connected to the nodes (ui, ui-1, ..., u1, ui+1, ..., un) with 2 ≤ i ≤ n.

For example, the pancakes of dimensions 2, 3, and 4 are shown in Figure 6.

****



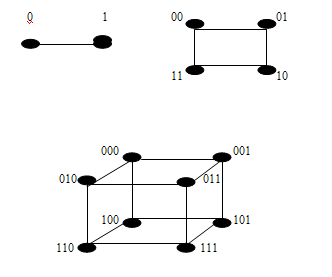
*G4*

**Exercise 07:**

*N*=2*d* nodes with d is a dimension of hypercube

*M*= d×2(*d*-1)) edges

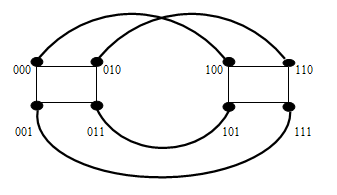
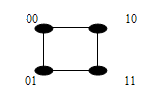
* A binary number of d-bits labels each node.
* Connections are established only between nodes whose numbers differ by just one bit. (Two nodes are connected only if their Hamming distance is equal to 1).

****

**Variation of hypercube named Crossed hypercube**

**Notation: Two binary strings x = x1x0 and y = y1y0 of length 2 are in a parity relation denoted (x ~ y) if and only if: (x, y) ∈ {(00, 00), (10, 10), (01, 11), (11, 01)}.**

* Each node of *CQn* prefixed by 0 is adjacent to a single node prefixed by 1 and vice versa. In the ordinary Hypercube, each edge is incident to two nodes u and v if and only if one label of u differs from the label of v by just one bit. However, in *CQn* (len-dimensional Crossed Cubes), the connections between two nodes are made according to the following rules:
  + For all n ≥ 1, (un-1un-2...u1u0, vn-1…v1v0) is an edge of *CQn* if and only if there exists an "*l*" such that:
    1. un-1…u*l* = vn-1…v*l*.
    2. u*l*-1 ≠ v*l*-1.
    3. ul-2 ≠ vl-2 if l is even.
    4. For 0 ≤ i ≤ ⌊(n-1)/2⌋, u2i+1u2i and v2i+1v2i are in a parity relation.



**CQ2 CQ3**

**Exercise (give during course)**

**Theorem**

Let G = (X; A) be a directed graph, with X = {x1, x2, …, xn}; of adjacency matrix M = (mi,j). For every natural integer k, not equal to zero, we denote Mk = (i,j). Then m(k) i,j is equal to the number of paths of length k from vertex xi to vertex xj.

**Proof**

We will perform induction on *k*: m*1*( i,j) represents the paths going from xi to xj. Assume the result is true for the integer *k - 1*; since Mk = Mk - 1 M; we have:



The number of paths in a graph. To clarify, you establish that the number of paths of length (k-1) going from (xi) to (xl) multiplied by (m{l,j}) (which indicates whether (xl, xj) is an edge) gives the number of paths of length (k) going from (xi) to (xj). Indeed, by using the induction hypothesis, you show that each path of length (k) can be constructed by taking a path of length(*k-1*) and adding an edge at the end. The sum of all these paths, for each (l) where (m\_{l,j} = 1), gives you the total number of paths of length (k) between (xi) and (xj). It's a beautiful way to use recursion to prove properties about graphs!