Graph and Network Theory

For a fixed $k \in \mathbb{N}$ let

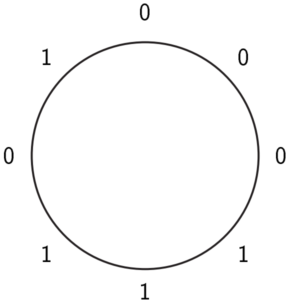
$$\mathcal{A}_k := \{0, 1, 2, ..., k-1\}.$$

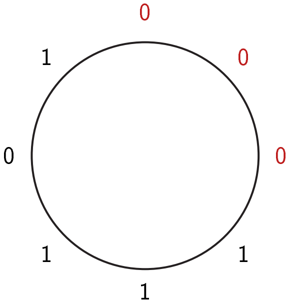
The set $\mathcal{A}_k$ will be called an **alphabet** with z k letters $0, 1, 2, ..., k-1$.

A **word** of length n over the alphabet $\mathcal{A}_k$ is a $n-$elements sequence with elements from $\mathcal{A}_k$.

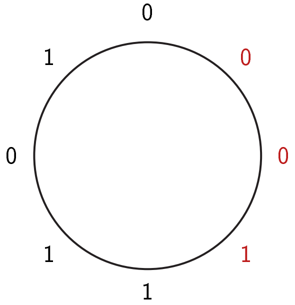
A **cyclic word** of length s over the alphabet $\mathcal{A}_k$ is any cyclic sequence of length s with elements from $\mathcal{A}_k$.

De Bruijn sequence of order n over the alphabet $\mathcal{A}_k$ is any cyclic word of length k^n which contains all possible words of length n.

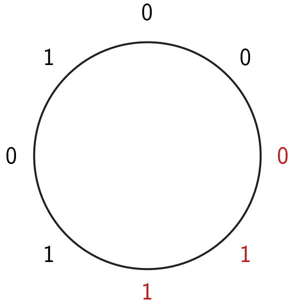

0
1
0
0
0
1
1
1



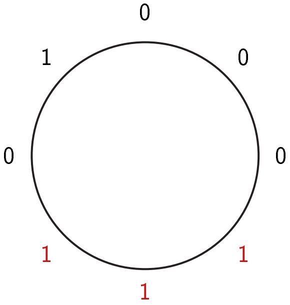
Moving around the circle, we can meet each of the $2^3 = 8$ binary strings of length 3: 000, 001, 011, 111, 110, 101, 010, 100 exactly one time.



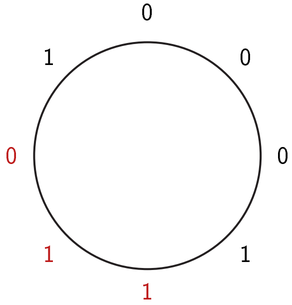
Moving around the circle, we can meet each of the $2^3 = 8$ binary strings of length 3: 000, 001, 011, 111, 110, 101, 010, 100 exactly one time.



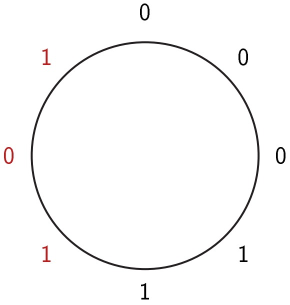
Moving around the circle, we can meet each of the $2^3 = 8$ binary strings of length 3: 000, 001, 011, 111, 110, 101, 010, 100 exactly one time.



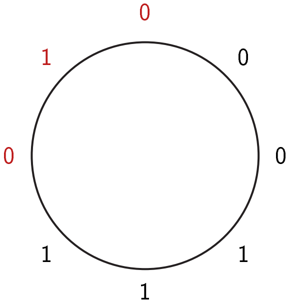
Moving around the circle, we can meet each of the $2^3 = 8$ binary strings of length 3: 000, 001, 011, 111, 110, 101, 010, 100 exactly one time.



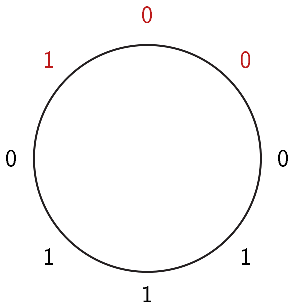
Moving around the circle, we can meet each of the $2^3 = 8$ binary strings of length 3: 000, 001, 011, 111, 110, 101, 010, 100 exactly one time.



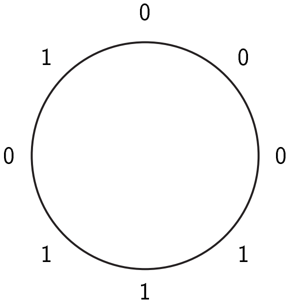
Moving around the circle, we can meet each of the $2^3 = 8$ binary strings of length 3: 000, 001, 011, 111, 110, 101, 010, 100 exactly one time.



Moving around the circle, we can meet each of the $2^3 = 8$ binary strings of length 3: 000, 001, 011, 111, 110, 101, 010, 100 exactly one time.



Moving around the circle, we can meet each of the $2^3 = 8$ binary strings of length 3: 000, 001, 011, 111, 110, 101, 010, 100 exactly one time.



Moving around the circle, we can meet each of the $2^3 = 8$ binary strings of length 3: 000, 001, 011, 111, 110, 101, 010, 100 exactly one time. **Thus, in an efficient way was encoded a binary string of 24 bits in length, only on 8 bits.**

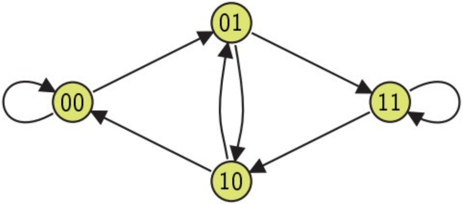
Examples

For the alphabet $A = \{0, 1\}$ ($k = 2$)

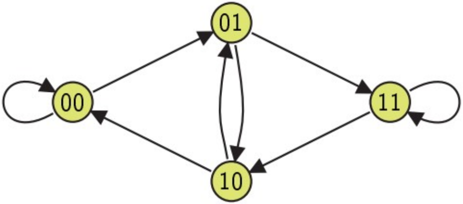
- for $n = 1$: 01
- for $n = 2$: 0110
- for $n = 3$: 01110100
- for $n = 4$: 0000100110101111

For the alphabet $A = \{0, 1, 2\}$) and $n = 3$: 011220210 *n*−**dimensional de Bruijn graph** $G_B(n,k)$ of the alphabet $\mathcal{A}_k$ is the directed graph $G = \langle V, E\rangle$, which vertices corresponds bijectively to n−lenght words, and

$$E = \{((v_1, v_2, ..., v_n), (v_2, v_3, ..., v_n, v)) : v_1, v_2, ..., v_n, v \in \mathcal{A}_k\}.$$



It is possible to assign wedges to the edges of de Bruijn graph. Namely, to the edge $e = ((v_1, v_2, ..., v_n), (v_2, v_3, ..., v_n, v))$ we assign v.



Properties of de Bruijn graphs

- two edges enter the vertex $a_1 a_2 \ldots a_{n-1} a_n$

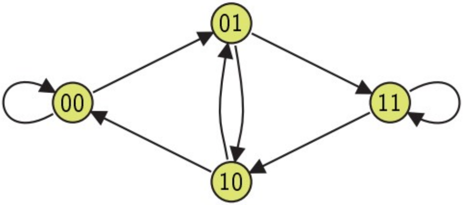
$$0a_1 a_2 \ldots a_{n-1} \mapsto a_1 a_2 \ldots a_n$$
$$1a_1 a_2 \ldots a_{n-1} \mapsto a_1 a_2 \ldots a_n$$

- there are two edges leaving the vertex $a_1 a_2 \ldots a_{n-1} a_n$

$$a_1 a_2 \ldots a_{n-1} a_n \mapsto a_1 a_2 \ldots a_{n-1} 0$$
$$a_1 a_2 \ldots a_{n-1} a_n \mapsto a_1 a_2 \ldots a_{n-1} 1$$

- for each vertex v we have (for $k = 2$)

$$degin(v) = degout(v) = 2$$



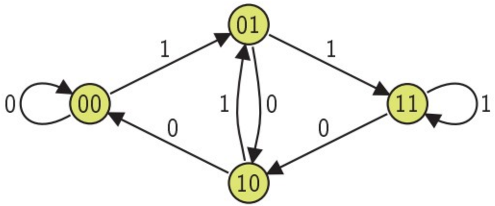
Properties of de Bruijn graphs

- the graph G_B is strongly connected
- path from the vertex $a_1 a_2 \ldots a_{n-1}$ to the vertex $b_1 b_2 \ldots b_{n-1}$ is generated as follows

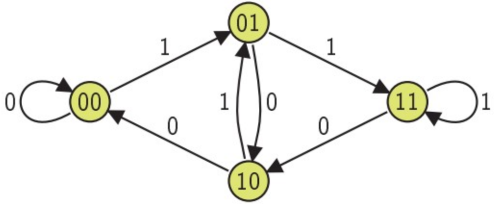
$$a_1 a_2 \ldots a_{n-1} b_1 b_2 \ldots b_{n-1}$$

$$(a_1 a_2 \ldots a_{n-1}), (a_2 \ldots a_{n-1} b_1), (a_3 \ldots a_{n-1} b_2 b_3), \ldots$$

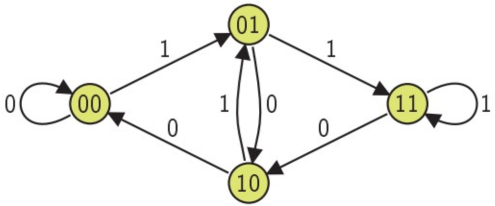
$$(a_{n-1} b_1 b_2 \ldots b_{n-2}), (b_1 b_2 \ldots b_{n-1} b_{n-1})$$



De Bruijn sequence of the order n over the alphabet $\mathcal{A}_k$ can be constructed by finding an **eulerian cycles in $(n-1)-$dimensional de Bruijn graph of the alphabet $\mathcal{A}_k$**.



De Bruijn sequence of the order n over the alphabet $\mathcal{A}_k$ can be constructed using **the Hamiltonian cycle in the de Bruijn's graph** $G_B(k, n)$.



De Bruijn sequence of order n = Eulerian cycle in $G_B(n-1,k)$ = Hamiltonian cycle in $G_B(n,k)$.

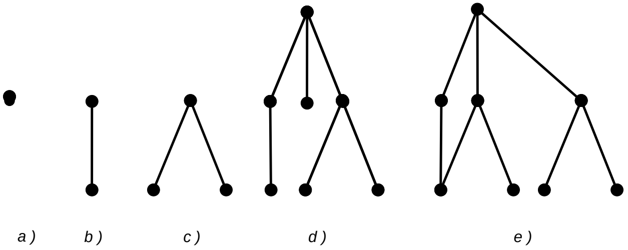
De Bruijn graphs

https://en.wikipedia.org/wiki/Peer-to-peer

A connected (undirected) graph without cycles is called a **tree**. An acyclic graph which is unconnected is called a **forest**.

By definition, it follows that the tree is simple graph, ie. there is no loops or multiple edges. Otherwise, it would have cycles, which contradicts the definition of a tree.

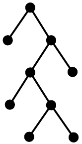
A vertex in a tree with the order of 1 is called a **leaf**.



Theorem

Let T be a graph witch n vertices. The following conditions are equivalent.

1. T is a tree,
2. T is an acyclic graph with $n-1$ edges,
3. T is a connected graph with $n-1$ edges,
4. T is connected, but is no longer connected after removing of any edge,
5. any two vertices of T are connected by exactly one path,
6. T is an acyclic graph, but after addition of any new edge the new graph has exactly one cycle.



A tree T, with at least two vertices $n \geq 2$, has at least two leafs.

Proof. Let T be any tree with $n \geq 2$ vertices and let w be the vertex with the smallest degree in the T tree.

Since T is a connected graph, so the vertex w has at least one neighbor, so the $deg(w) \geq 1$.

Suppose that all vertices except the vertex w have degree greater than or equal to 2. Then we have

$$2|E_T| = \sum_{v \in V} deg(v) \geq 2(n-1) + 1$$

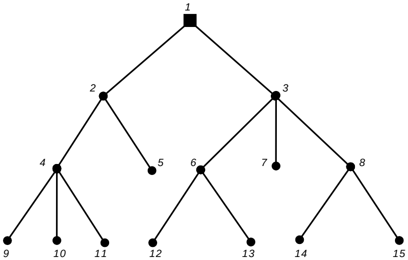
On the other hand, T is a tree, so $|E_T| = n - 1$.

Hence we have a contradiction.

Thus, there must exist a vertex $x \neq w$, which has the degree $deg(x) = 1$. Since the vertex w has the smallest degree, so $deg(w) = 1$. ∎

Trees with a root

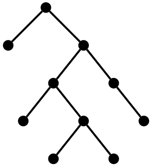
In many cases the tree which we use to solve practical problems has hierarchical structure, i.e. it has one distinguished vertex called a **root**.



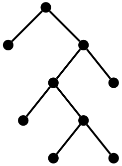
In this tree, the leaves are called external nodes or terminal nodes. Other nodes in the tree are called internal nodes..

- In a rooted tree a **parent** p of a vertex v is the vertex connected to v on the path from p to the root. A **child** of a vertex w is a vertex of which w is the parent.
- Each node (except the root) has exactly one parent.
- A parent may have several children.
- In general: w is a descendant of v if $w \neq v$ and the vertex v belongs to the simple path from w to the root.

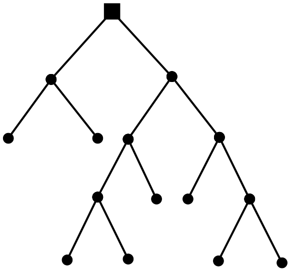
A **binary tree** is called a tree which all nodes are of degree at most 3 (every node has at most two children).



A **regular binary tree** is a tree in which every node has either 0 or 2 children.



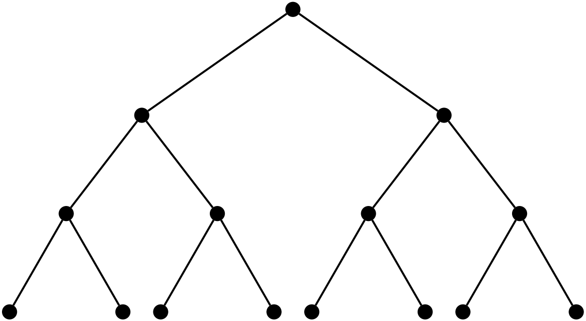
We say, that a node v in binary tree has the **level** of d if it's distance from the root is d.



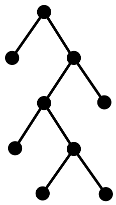
The height of tree is $l_{max} = 4$

The maximum level l_{max} of a node in binary tree is called the **height of tree**.

A **complete binary tree** is a regular binary tree in which all leaves has the same level (equal the height of the tree).



The number of vertices n in a regular binary tree is always **odd**



Proof. By the definition of a regular binary tree, we know that only one vertex (root) is of even (second) degree the remaining $n-1$ vertices are of odd degree (first or third).

On the other hand, we know that in any graph (and therefore in a tree) the number of vertices of odd degree is even, i.e. $n-1$ is an even number, so n is an odd number. ∎

Let p be the number of leaves in a regular binary tree with n vertices. Then

$$p = \frac{n+1}{2}$$

Proof. Let's notice that $n - p - 1$ is the number of vertices of the of degree three. Hence, from the **handshake lemma**, the number of edges $|E|$ in the tree is

$$2|E| = 1 \cdot p + 2 \cdot 1 + 3 \cdot (n - p - 1)$$

From the properties of trees we have

$$|E| = n - 1$$

So

$$p + 2 + 3n - 3p - 3 = 2(n-1)$$
$$3n - 2p - 1 = 2n - 2$$
$$2p = n + 1$$
$$p = \frac{n+1}{2}$$

■

Properties of binary trees - the number of internal nodes in the regular binary tree

A regular binary tree with n vertices has

$$w = \frac{E_T}{2}$$

internal nodes.

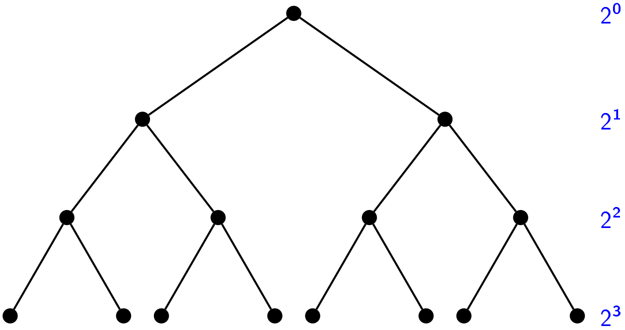
Proof. Let's notice that if p is the number of leaves in a binary tree, then $p = \frac{n+1}{2}$. Therefore, $n - p$ is the number of of internal nodes in the binary tree.

$$n - p = n - \frac{n+1}{2} = \frac{n-1}{2}$$

■

The maximum number of vertices in a regular binary tree with k levels is equal

$$n_{max} = 2^0 + 2^1 + 2^2 + ... + 2^k = 2^{k+1} - 1$$



$$n_{max} = 2^0 + 2^1 + 2^2 + 2^3 = 1 + 2 + 4 + 8 = 15$$

What can be the minimum height of a regular binary tree having n vertices?

What can be the minimum height of a regular binary tree having n vertices?

- at level 0 there can be only one vertex (root)

What can be the minimum height of a regular binary tree having n vertices?

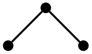
- at level 0 there can be only one vertex (root)

What can be the minimum height of a regular binary tree having n vertices?

- at level 0 there can be only one vertex (root)
- at level one - two vertices

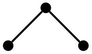
What can be the minimum height of a regular binary tree having n vertices?

- at level 0 there can be only one vertex (root)
- at level one - two vertices



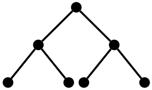
What can be the minimum height of a regular binary tree having n vertices?

- at level 0 there can be only one vertex (root)
- at level one - two vertices
- on the second level - a maximum of four vertices



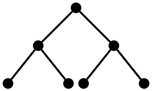
What can be the minimum height of a regular binary tree having n vertices?

- at level 0 there can be only one vertex (root)
- at level one - two vertices
- on the second level - a maximum of four vertices



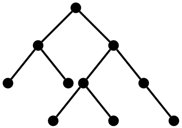
What can be the minimum height of a regular binary tree having n vertices?

- at level 0 there can be only one vertex (root)
- at level one - two vertices
- on the second level - a maximum of four vertices
- at the $k-$th level a maximum of 2^k vertices



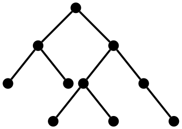
What can be the minimum height of a regular binary tree having n vertices?

- at level 0 there can be only one vertex (root)
- at level one - two vertices
- on the second level - a maximum of four vertices
- at the $k-$th level a maximum of 2^k vertices



What can be the minimum height of a regular binary tree having n vertices?

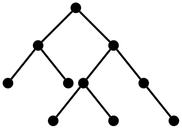
- at level 0 there can be only one vertex (root)
- at level one - two vertices
- on the second level - a maximum of four vertices
- at the $k-$th level a maximum of 2^k vertices



$$2^0 + 2^1 + 2^2 + ... + 2^k \geq n$$

What can be the minimum height of a regular binary tree having n vertices?

- at level 0 there can be only one vertex (root)
- at level one - two vertices
- on the second level - a maximum of four vertices
- at the $k-$th level a maximum of 2^k vertices

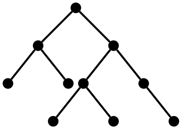


$$2^0 + 2^1 + 2^2 + ... + 2^k \geq n$$

$$2^{k+1} - 1 \geq n$$

What can be the minimum height of a regular binary tree having n vertices?

- at level 0 there can be only one vertex (root)
- at level one - two vertices
- on the second level - a maximum of four vertices
- at the $k-$th level a maximum of 2^k vertices



$$2^0 + 2^1 + 2^2 + \ldots + 2^k \geq n$$

$$2^{k+1} - 1 \geq n$$

$$2^{k+1} \geq n + 1$$

$$k + 1 \geq \log_2(n+1)$$

$$\min l_{max} = \lceil \log_2(n+1) - 1 \rceil$$

What can be the maximum height of a regular binary tree having n vertices?

What can be the maximum height of a regular binary tree having n vertices?

- at level 0 there can be only one vertex (root).

What can be the maximum height of a regular binary tree having n vertices?

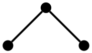
- at level 0 there can be only one vertex (root).

What can be the maximum height of a regular binary tree having n vertices?

- at level 0 there can be only one vertex (root).
- at level one - two vertices.

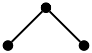
What can be the maximum height of a regular binary tree having n vertices?

- at level 0 there can be only one vertex (root).
- at level one - two vertices.



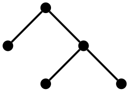
What can be the maximum height of a regular binary tree having n vertices?

- at level 0 there can be only one vertex (root).
- at level one - two vertices.
- at level two - a maximum of two vertices.



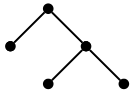
What can be the maximum height of a regular binary tree having n vertices?

- at level 0 there can be only one vertex (root).
- at level one - two vertices.
- at level two - a maximum of two vertices.



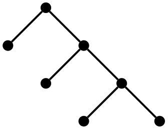
What can be the maximum height of a regular binary tree having n vertices?

- at level 0 there can be only one vertex (root).
- at level one - two vertices.
- at level two - a maximum of two vertices.
- at level $k-$th maximum 2 vertices.



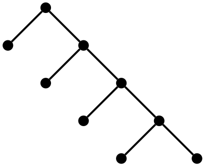
What can be the maximum height of a regular binary tree having n vertices?

- at level 0 there can be only one vertex (root).
- at level one - two vertices.
- at level two - a maximum of two vertices.
- at level $k-$th maximum 2 vertices.



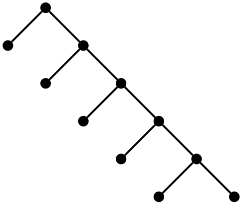
What can be the maximum height of a regular binary tree having n vertices?

- at level 0 there can be only one vertex (root).
- at level one - two vertices.
- at level two - a maximum of two vertices.
- at level $k-$th maximum 2 vertices.



What can be the maximum height of a regular binary tree having n vertices?

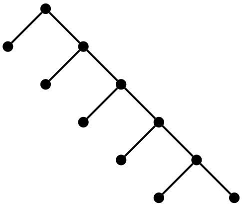
- at level 0 there can be only one vertex (root).
- at level one - two vertices.
- at level two - a maximum of two vertices.
- at level $k-$th maximum 2 vertices.



Search problem

What can be the maximum height of a regular binary tree having n vertices?

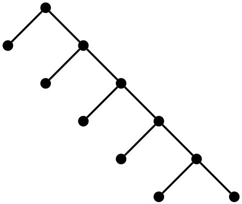
- at level 0 there can be only one vertex (root).
- at level one - two vertices.
- at level two - a maximum of two vertices.
- at level $k-$th maximum 2 vertices.



$$
\begin{aligned}
n &= 2k+1 \\
\max l_{max} &= \frac{n-1}{2}
\end{aligned}
$$

What can be the maximum height of a regular binary tree having n vertices?

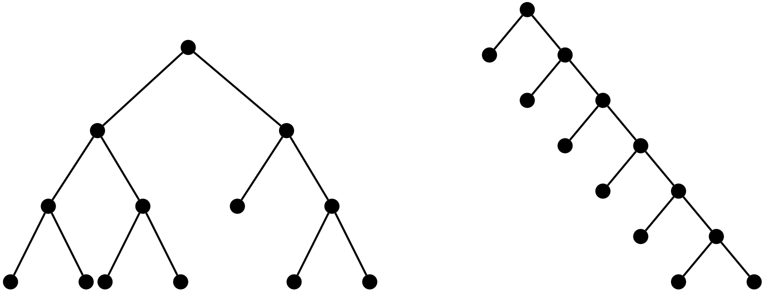
- at level 0 there can be only one vertex (root).
- at level one - two vertices.
- at level two - a maximum of two vertices.
- at level $k-$th maximum 2 vertices.



$$\begin{aligned} n &= 2k+1 \\ \max l_{max} &= \frac{n-1}{2} \end{aligned}$$

The search problem - an example for $n = 13$

1. $\min l_{max} = \lceil \log_2(n+1) - 1 \rceil = \lceil \log_2(14) - 1 \rceil = 3$
2. $\max l_{max} = \frac{n-1}{2} = \frac{13-1}{2} = 6$



for $n = 30001$, we have $\min l_{max} = 15$ i $\max l_{max} = 15000$


HAPPY
EASTER

Thank you for your attention!!!