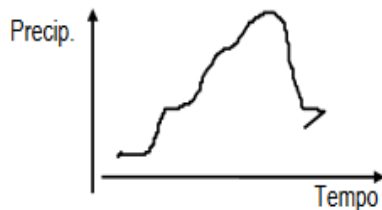


1. Introdução

Considere que existe uma coleção de dados representando as condições diárias da precipitação pluviométrica num determinado período do mês. Se o desejado é a média mensal desta precipitação basta somar todos os dados e dividi-los pelo número de dias do mês. Porém se quisermos traçar um gráfico representativo da função precipitação no tempo, necessitaremos ordenar uma coleção de dados seguindo a ordem cronológica:



Esta forma de estrutura de dados, na qual o tipo de relacionamento entre os dados é a relação de ordem, é a mais simples e comum e chamamos de ordenada ou linear.

São exemplos de estruturas lineares:

- dias da semana,
- cartas de baralho,
- pauta de chamada,
- anos em que um time foi campeão

Exemplo:

```
sinonimos_dict = {'bom': ['agradavel', 'otimo']}  
altera_sinonimos("hoje o dia esta bom")
```

Out: "hoje o dia esta otimo"

2. Definição

Uma lista linear é uma estrutura dinâmica caracterizada por uma sequência ordenada de elementos, chamados de nós, no sentido da sua posição relativa: x_1, x_2, \dots, x_n , tal que:

1. Existem n nós na sequência ;
2. x_1 é o primeiro nó da sequência;
3. x_n é o último nó da sequência;
4. Para todo i, j entre 1 e n , se $i < j$, então o nó x_i antecede o nó x_j ;
5. Caso $i = j - 1$, x_i é o antecessor de x_j e x_j é o sucessor de x_i .

Existem várias operações que podem ser realizadas em listas lineares. Por exemplo, em uma lista de alunos inscritos em um curso podemos verificar a presença dos alunos (Percurso), novos alunos são matriculados (inserção) e alguns alunos, eventualmente, desistem (retirada).

Portanto:

- Uma coleção de objetos do mesmo tipo que possuem um relacionamento posicional entre si é chamada de Lista Linear.
- Uma Lista Linear é composta de nós que contém as informações sobre os objetos. Estes nós podem ser de um tipo primitivo ou definido pelo programador. Assim, definimos uma Lista Linear como sendo o conjunto de $n \geq 0$ nós denominados $x_1, x_2, \dots, x_k, \dots, x_n$ organizados estruturalmente de forma a refletir as posições relativas dos mesmos:
 - Se $n > 0$ então x_1 é o primeiro nó e x_n é o último nó;
 - Para $1 < k < n$, o nó x_k é precedido pelo nó x_{k-1} chamado antecessor e sucedido pelo nó x_{k+1} , chamado sucessor;
 - Quando $n = 0$, dizemos que a lista está vazia.
 -

3. Propriedades de uma Lista

1. Uma Lista pode ter 0 ou mais nós;
2. Um nó pode ser incluído em qualquer posição da lista;
3. Qualquer nó da lista pode ser eliminado;
4. Qualquer nó da lista pode ser acessado;
5. Podemos visitar cada nó da lista na sua vez.

4. Operações básicas em uma lista

- **Percurso** : operação que permite utilizar cada um dos nós de uma lista linear de tal forma que:
 - O primeiro nó a ser utilizado é o primeiro nó da lista linear
 - Para utilizar um nó x_j , todos os nós de x_1 até x_{j-1} já foram utilizados
 - O último nó utilizado é o último nó da lista linear.
- **Acesso a um nó**: operação que procura um nó específico da lista linear. A busca pode ser efetuada de duas formas: ou o nó é identificado por sua posição relativa na lista ou o nó é identificado pelo seu conteúdo. O objetivo deste acesso pode ser tanto para consultar informações como para alterá-las.
- **Inserção** : operação que incorpora um nó x a uma lista linear, de tal forma que :
 - O nó x terá um sucessor e/ou um antecessor;
 - A inserção é realizada de tal forma que ao inserir o nó x na posição i , onde i está no intervalo de 1 a $(n+1)$, esse nó passa a ser o i -ésimo nó da lista linear;
 - O número de nós (n) é acrescido de uma unidade.
- **Retirada**: operação que retira um nó X de uma lista linear, de tal forma que :
 - Se x_i é o nó retirado, o seu sucessor passa a ser o sucessor de seu antecessor, ou seja, x_{i+1} passa a ser o sucessor de x_{i-1} . Se x_i é o primeiro nó, o seu sucessor passa a ser o primeiro; Se x_i é o último nó, o seu antecessor passa a ser o último;
 - O número de nós (n) é decrescido de uma unidade.

Nota: Nos casos de *Busca*, *Inserção* e *Retirada*:

1. se a lista tem seus nós posicionados de tal forma que há classificação por um campo de informação do nó, chamado chave, tal classificação deve ser preservada.

2. se a lista não permite que haja dois nós com mesmo valor em um campo de informação, chamado chave, tal situação(não repetição) deve ser preservada