[LAB] Pseudocodes and flowcharts

1. Snail and well:

Objetivo: Saber quantos dias leva para o caramujo escapar do poço.

Valores que devem ser considerados:

- Altura do poço;

- Distância que o caramujo avança por dia;

- Regresso do caramujo durante à noite;

- Distância acumulada: diferença entre o avanço e o regresso;

- Quantidade de dias (o que deve ser calculado)

Observação: para o último dia, em que o caramujo sobe, atinge a altura do poço, e não regressa, considerou-se que a nova altura do poço é (altura do poço – avanço):

Enquanto a distância acumulada for menor que a nova altura do poço:

Soma-se 1 dia na contagem;

Esta contagem deve parar quando a distância acumulada multiplicada pela quantidade de dias atingir o valor da nova altura do poço.

Mostre a quantidade de dias total.

1.1 Bônus:

Objetivo 1: dada uma lista de valores de avanço diários, defina quantos dias o caramujo leva para sair do poço.

Para cada elemento da lista dada:

Some este elemento ao elemento seguinte da lista;

Esta soma deve parar quando o resultado atingir valor maior ou igual à altura do poço.

Sendo assim, mostre a quantidade de dias. A quantidade de dias será definida pela posição do último número somado na lista (index).

Objetivo 2: calcular os deslocamentos máximo e mínimo em um dia.

Utilizar as funções max(lista) e min(lista) para este caso.

Objetivo 3: calcular a velocidade média do caramujo.

Velocidade = deslocamento / tempo

O deslocamento é definido pelos valores dos elementos da lista; o tempo, pela quantidade de elementos na lista.

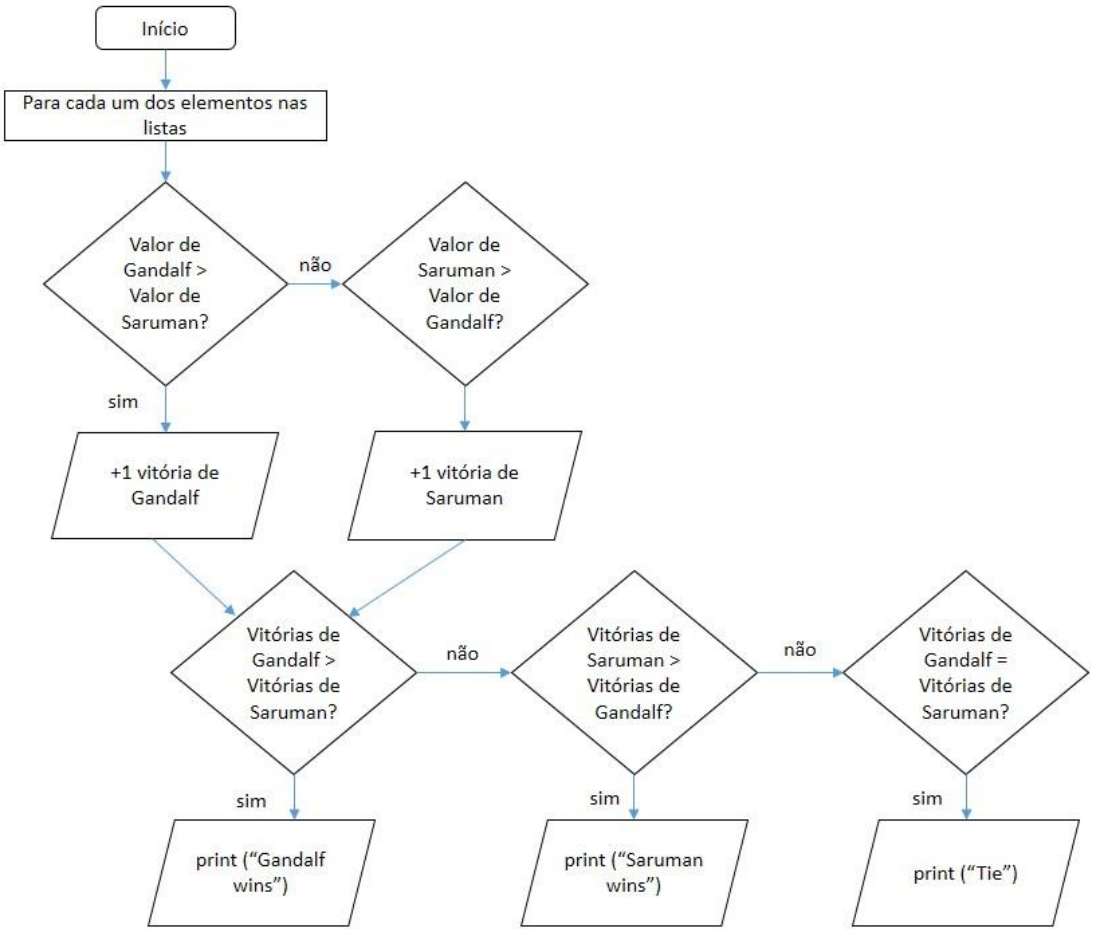
Então, a velocidade é calculada pela média da lista.

Objetivo 4: calcular o desvio padrão do deslocamento:

Importar o pacote statistics, e utilizar a função stdev(lista) para este caso.

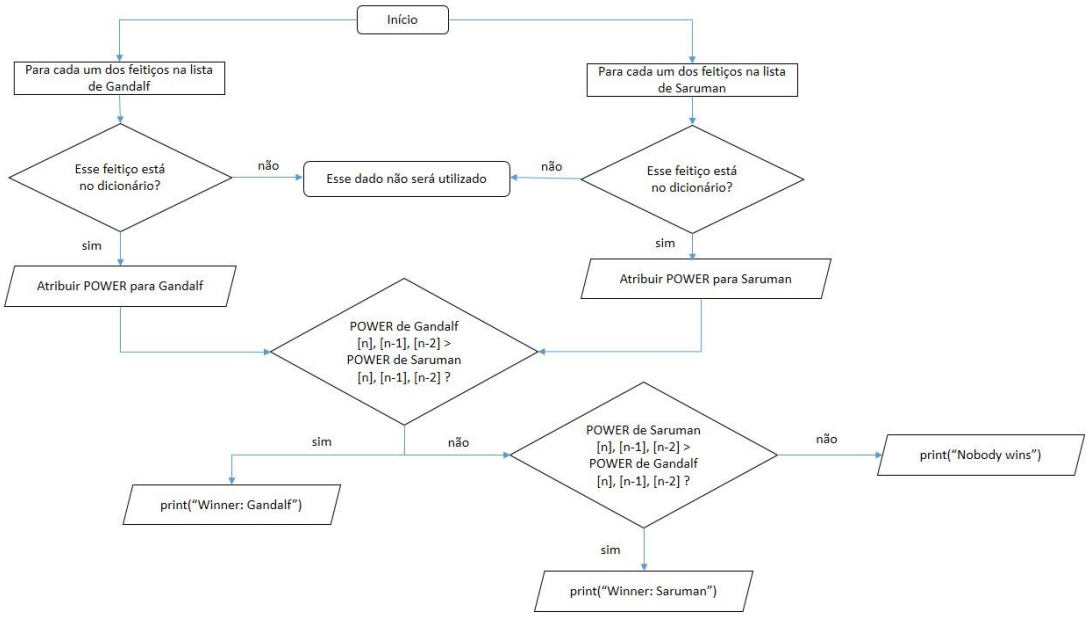
2. Duel of sorceres:

Objetivo: Comparar as listas de Gandalf e Saruman, para determinar quem ganhou mais batalhas.



2.1 Bônus:

Objetivo 1: dada as listas de feitiços, e seus respectivos valores relacionados no dicionário, verificar o ganhador. O ganhador será aquele que ganhar 3 batalhas consecutivas.



Objetivo 2: média de cada lista de feitiços.

A média de cada uma listas é dada pela soma dos valores que compõem a lista, dividida pela quantidade de elementos da lista.

Objetivo 3: desvio padrão de cada lista de feitiços.

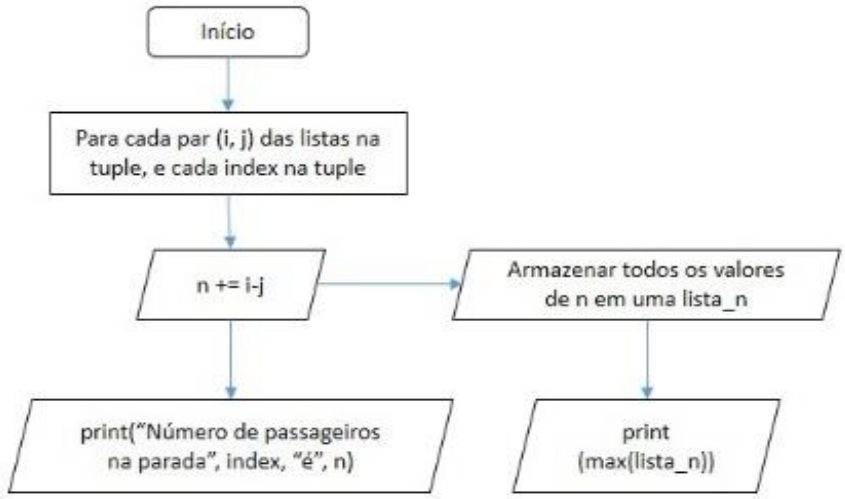
Importar o pacote statistics, e utilizar a função stdev(lista) para cada uma das listas.

3. Bus:

Objetivo 1: Calcular o número de paradas do ônibus.

Cada parada do ônibus é representada por uma lista, e tais listas estão reunidas em uma tuple. Portanto, para saber o número de paradas, deve-se calcular a length(tuple).

Objetivos 2 e 3: determinar o número de passageiros em cada parada, e a ocupação máxima do ônibus



Objetivo 4: média da ocupação do ônibus.

A média de cada uma listas é dada pela soma dos valores que compõem a lista, dividida pela quantidade de elementos da lista.

Objetivo 5: desvio padrão da ocupação do ônibus.

Importar o pacote statistics, e utilizar a função stdev(lista_n) para cada uma das listas.

4. Robin Hood:

Objetivo 1: Verificar se houve mais de uma flecha que acertou o mesmo lugar no alvo.

Calcular o len(tuple);

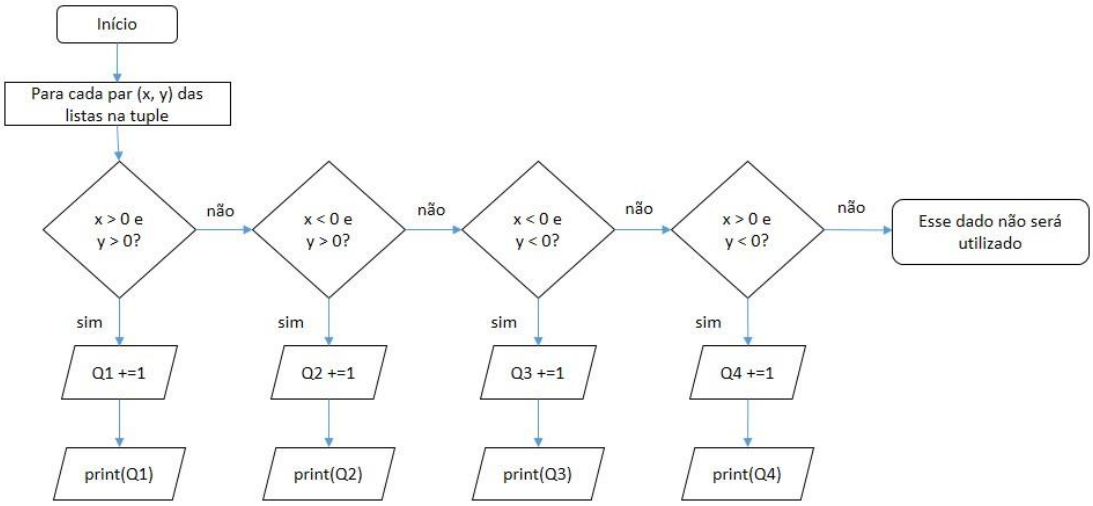
Transformar a tuple em set;

Calcular o len(set);

Calcular a diferença entre len(tuple) e len(set);

Mostre esse resultado.

Objetivo 2: Calcular quantas flechas caíram em cada quadrante.


Início
Para cada par (x, y) das listas na tuple
x > 0 e y > 0?
não
x < 0 e y > 0?
não
x < 0 e y < 0?
não
x > 0 e y < 0?
não
Esse dado não será utilizado
sim
Q1 +=1
print(Q1)
sim
Q2 +=1
print(Q2)
sim
Q3 +=1
print(Q3)
sim
Q4 +=1
print(Q4)