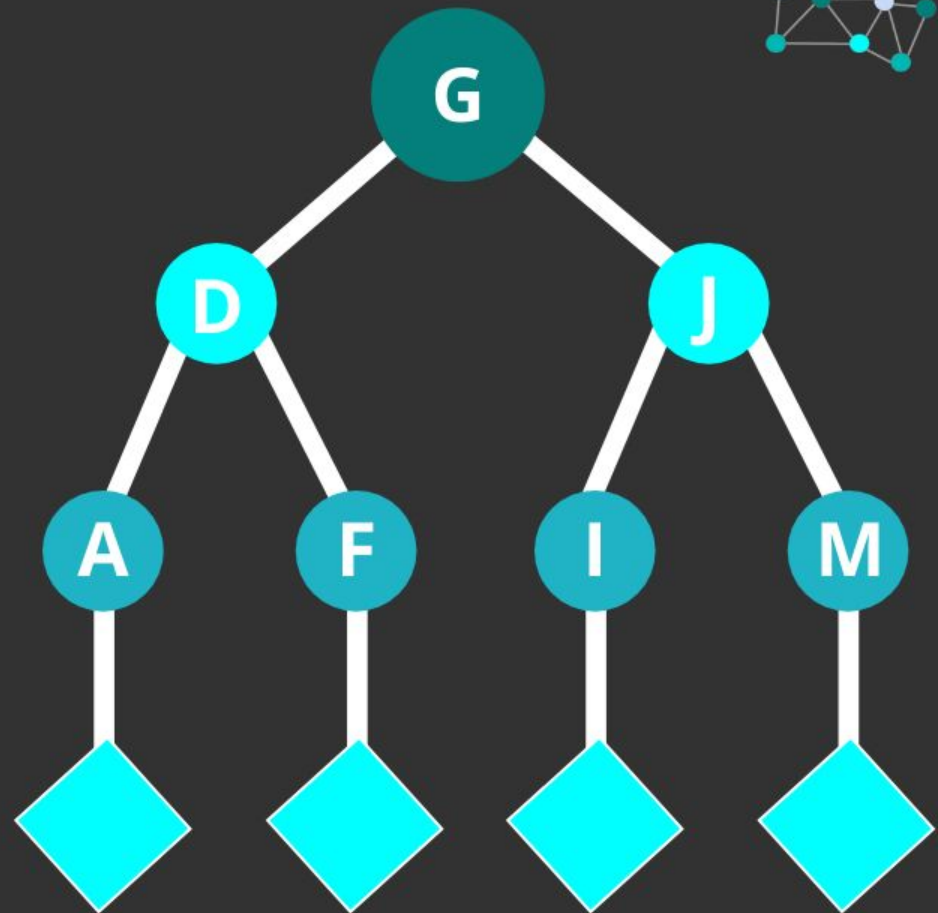
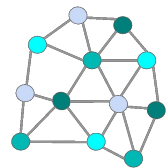


Leonardo de  
Abreu  
Schmidt

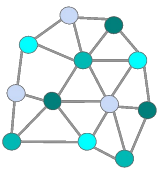


# Alocação



- Quando criamos uma variável basicamente reservamos (**alocamos**) um local na memória RAM para essa variável.
- Cada tipo de dado demanda um determinado **espaço** para que a variável seja colocada em memória.

# Alocação em C



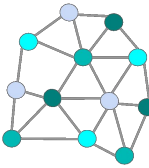
char: 1 byte

int: 4 bytes

float: 4 bytes

double: 8 bytes

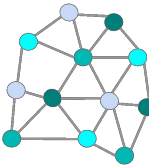
# Alocação em memória RAM



- Como um programa é alocado em memória?
- Como variáveis são alocadas em memória?
- Dados são todos alocados continuamente?
- Alocação é estática ou dinâmica?

Em linguagens de **alto nível** utilizamos estruturas e **não nos importamos** de que forma nossos dados são armazenados fisicamente.

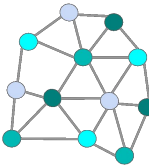
# Alocação em memória RAM



- **Como um programa é alocado em memória?**
- Como variáveis são alocadas em memória?
- Dados são todos alocados continuamente?
- Alocação é estática ou dinâmica?

Em linguagens de **alto nível** utilizamos estruturas e **não nos importamos** de que forma nossos dados são armazenados fisicamente.

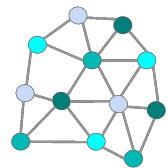
# Alocação em memória RAM



- Como um programa é alocado em memória?
- **Como variáveis são alocadas em memória?**
- Dados são todos alocados continuamente?
- Alocação é estática ou dinâmica?

Em linguagens de **alto nível** utilizamos estruturas e **não nos importamos** de que forma nossos dados são armazenados fisicamente.

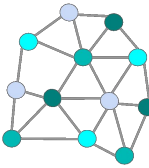
# Alocação em memória RAM



- Como um programa é alocado em memória?
- Como variáveis são alocadas em memória?
- **Dados são todos alocados continuamente?**
- Alocação é estática ou dinâmica?

Em linguagens de **alto nível** utilizamos estruturas e **não nos importamos** de que forma nossos dados são armazenados fisicamente.

# Alocação em memória RAM

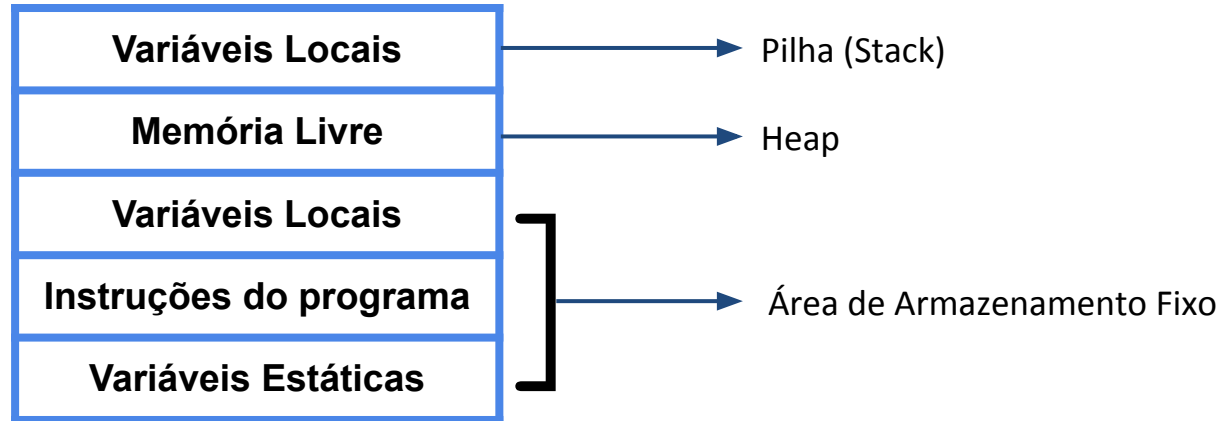
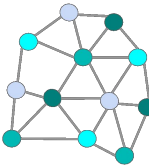


- Como um programa é alocado em memória?
- Como variáveis são alocadas em memória?
- Dados são todos alocados continuamente?
- **Alocação é estática ou dinâmica?**

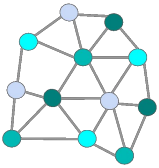
Em linguagens de **mais alto nível** utilizamos estruturas e **não nos importamos** de que forma nossos dados são armazenados fisicamente.



# Alocação de memória RAM



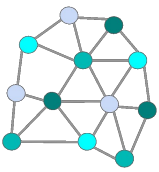
# Alocação em memória



O que é melhor?

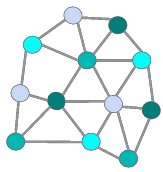
- Alocar continuamente todos os dados do meu programa?
- Ou alocar dinamicamente conforme demanda?

# Linearidade x Não Linearidade



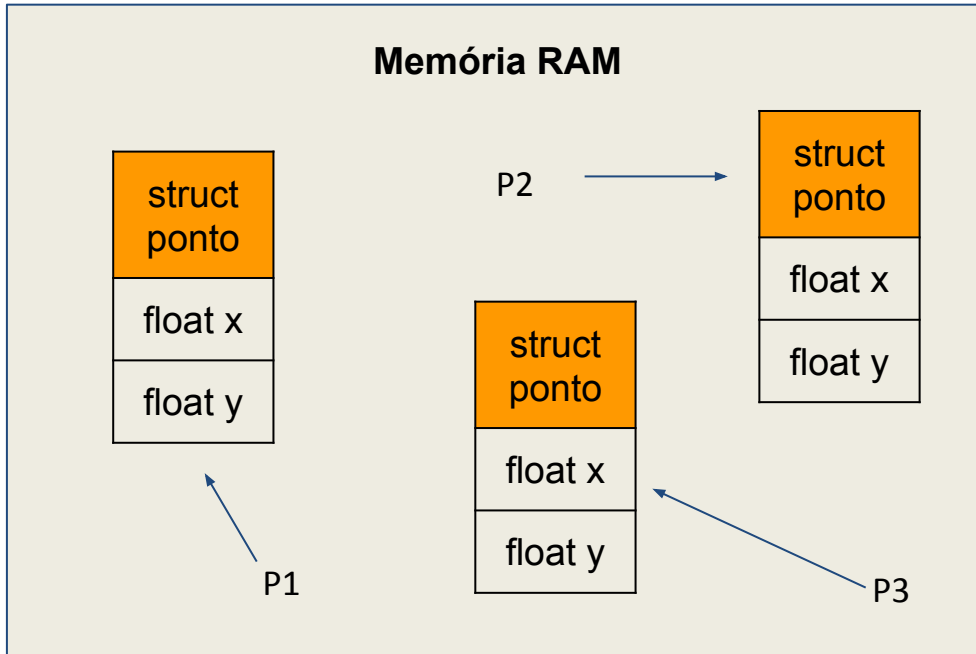
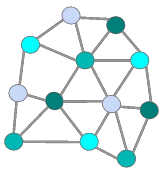
- **Estruturas Lineares:** são aquelas em que elementos estão alocados lado a lado na memória RAM. Ex: arrays, vetores.
- **Estruturas não-lineares:** são aquelas em que os elementos estão espalhados na memória, ou seja, não estão sequenciais. Então como saber onde elas estão?

# Estruturas não lineares



- Para saber onde essas estruturas estão podemos guardar um **ponteiro** que aponte para elas.
- Cada estrutura portanto tem um **ponteiro** que aponta para si na memória RAM.

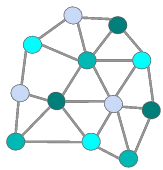
# Estruturas não lineares



Os pontos  $P_i$  são os ponteiros que apontam para os endereços de memória onde tem as structs de ponto.

Parece estranho né? **Mas então por que precisamos de ponteiros apontando para estruturas?**

# Estruturas não lineares

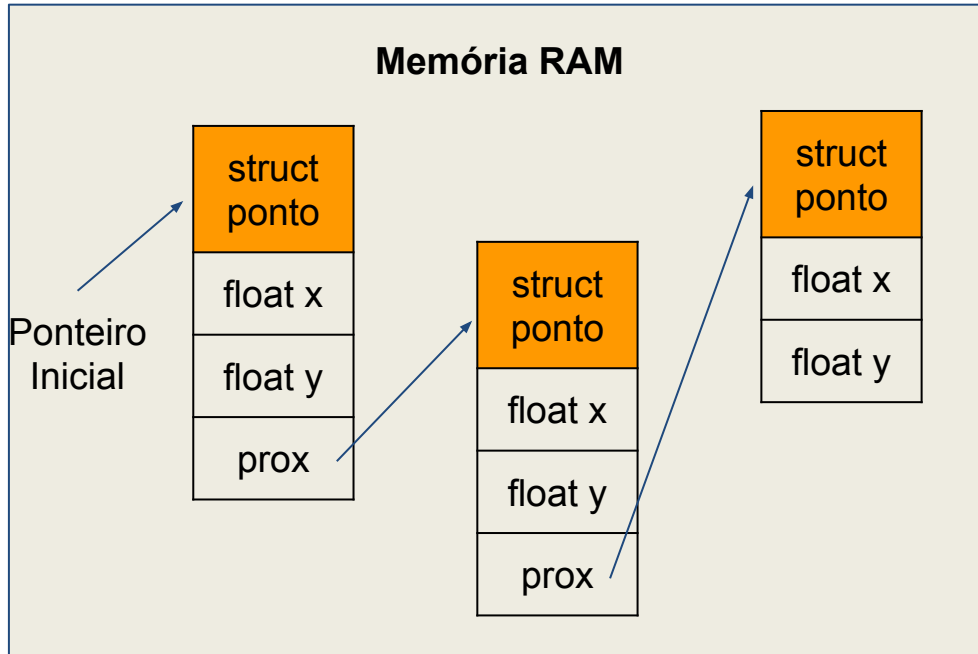
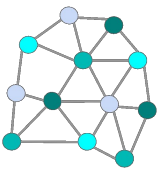


Bom a resposta da pergunta é que precisamos de ponteiros e structs para criar estruturas de dados não lineares.

Ex: listas, pilhas, filas, árvores, entre outras.

Além disso cada estrutura dessa tem variações para criar estruturas semelhantes e otimizadas.

# Estruturas não-lineares



A solução encontrada é **adicionar ponteiros em cada estrutura** juntamente com as variáveis  $x$  e  $y$  já existentes, apontando para a struct ponto seguinte na ordem em que são adicionadas.

Leonardo de  
Abreu  
Schmidt

