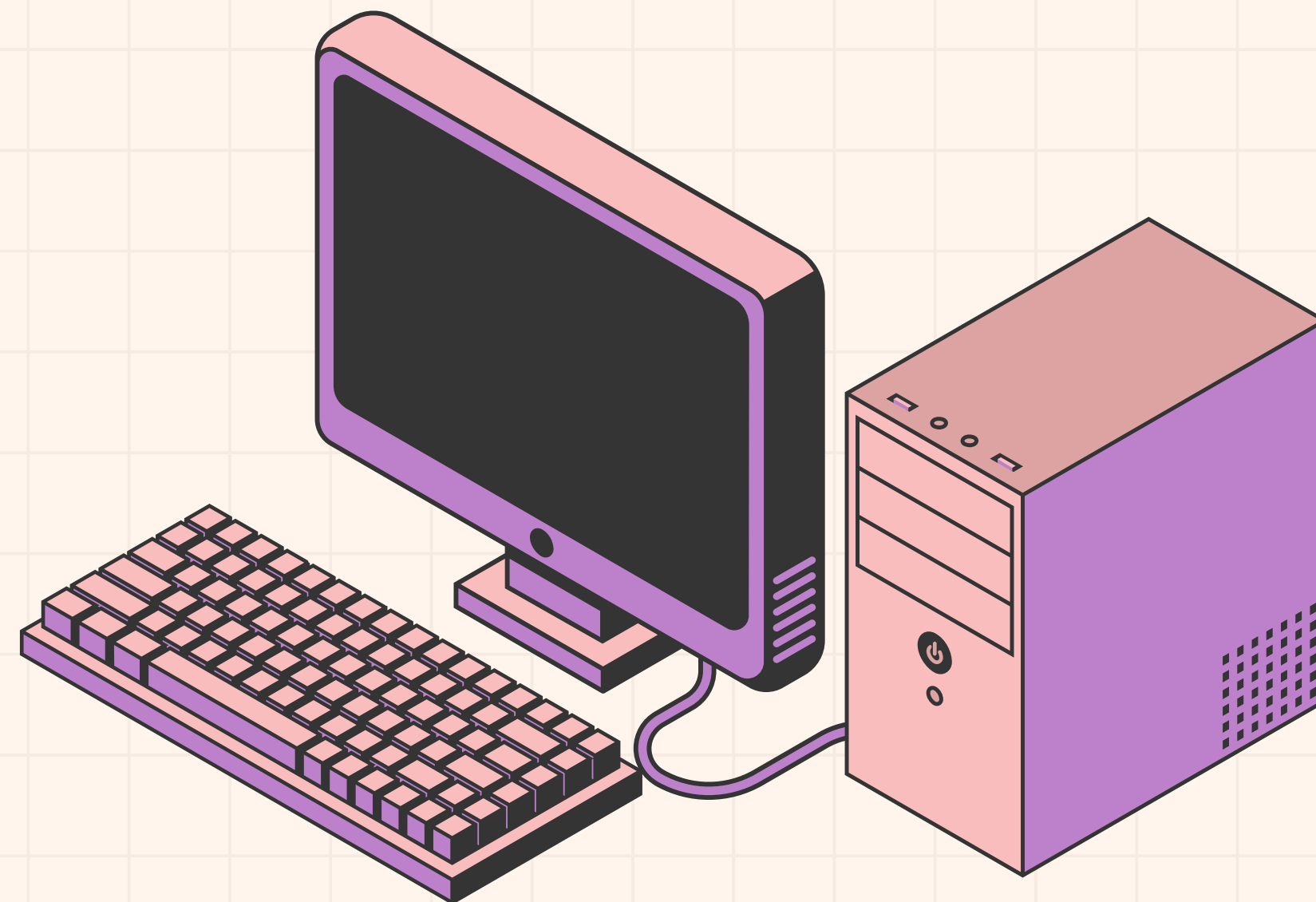


COURSE PROJECT

# PRIORITY QUEUE USING MIN/MAX HEAP

DATA STRUCTURES AND ALGORITHMS

CSC10004

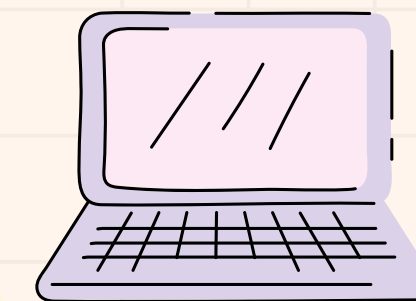


Trần Minh Hiếu Học - 23120006

Võ Quốc Gia - 23120014

Phan Ngưng - 23120304

# Nội dung



## 01

Mở đầu



## 02

Lịch sử hình  
thành Priority  
Queue



## 03

Một số định  
nghĩa - khái niệm



## 04

Priority Queue

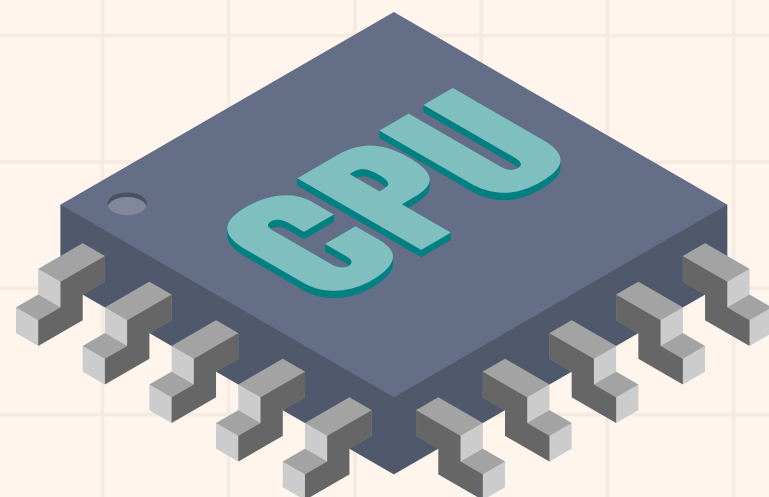


# 1. MỞ ĐẦU

Cấu trúc dữ liệu đóng vai trò quan trọng trong việc tối ưu hóa hiệu suất của các thuật toán và ứng dụng.

Priority Queue (Hàng đợi ưu tiên) là:

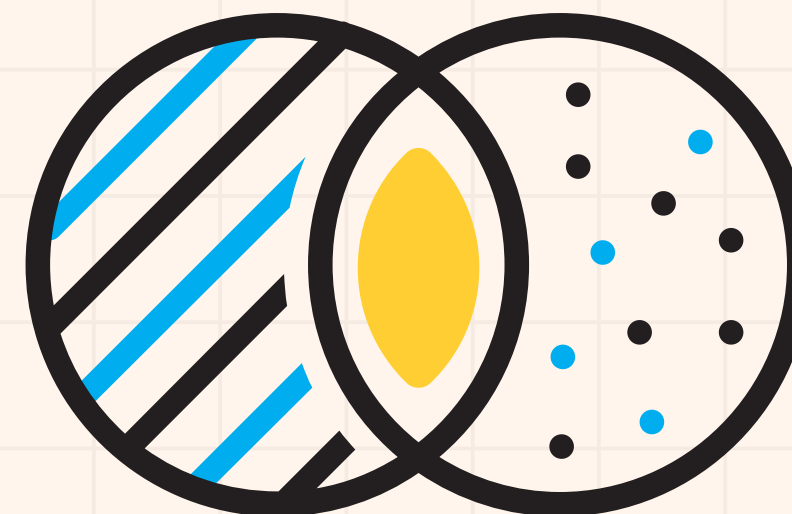
- Một trong những cấu trúc dữ liệu phổ biến
- Được sử dụng rộng rãi trong:



Lập lịch CPU



Thuật toán tìm đường  
đi ngắn nhất: DIJKSTRA

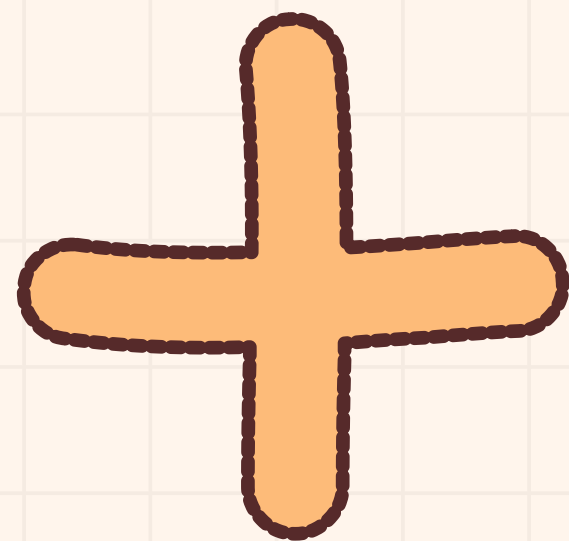


Các bài toán  
tổ hợp khác

# 1. MỞ ĐẦU

Bài báo này, ta sẽ tập trung nghiên cứu Binary Heap, một cấu trúc dữ liệu đặc biệt giúp triển khai hiệu quả Priority Queue.

Các thao tác cơ bản của Binary Heap gồm:



chèn phần tử  
(insert)



lấy phần tử có  
độ ưu tiên cao nhất  
(extract-min/max)

## 2. LỊCH SỬ HÌNH THÀNH CỦA PRIORITY QUEUE



John William Joseph Williams  
(9/1930 - 29/9/2012)

### ALGORITHM 232 HEAPSORT

J. W. J. WILLIAMS (Recd 1 Oct. 1963 and, revised, 15 Feb. 1964)

Elliott Bros. (London) Ltd., Borehamwood, Herts, England

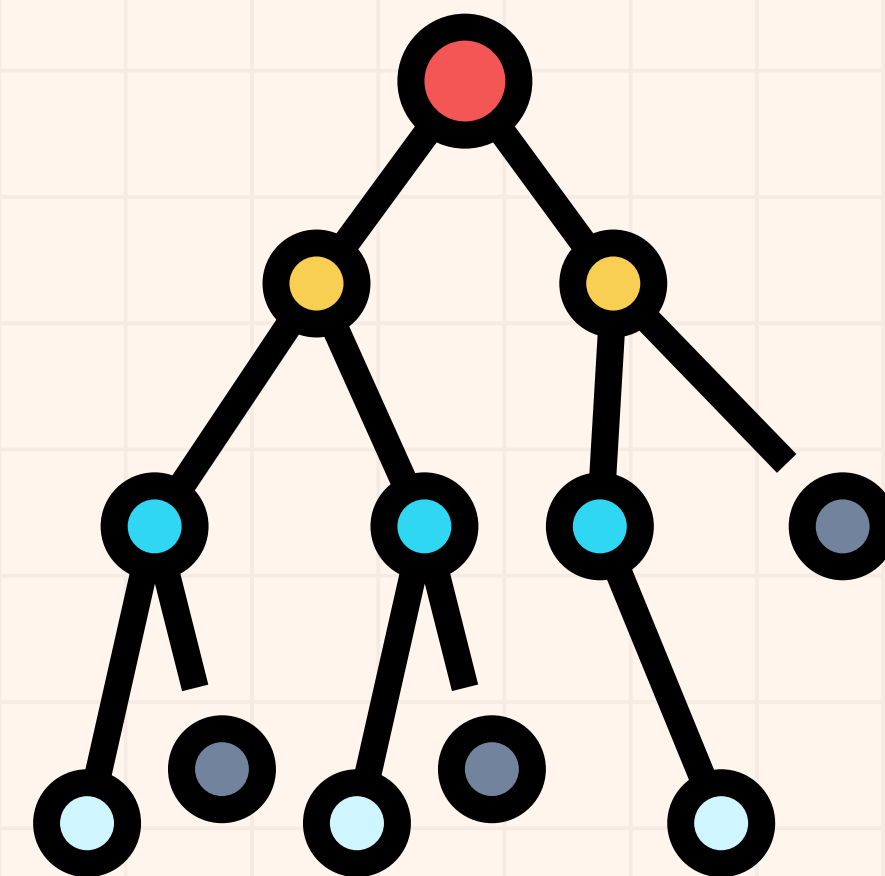
**comment** The following procedures are related to *TREESORT* [R. W. Floyd, Alg. 113, *Comm. ACM* 5 (Aug. 1962), 434, and A. F. Kaupé, Jr., Alg. 143 and 144, *Comm. ACM* 5 (Dec. 1962), 604] but avoid the use of pointers and so preserve storage space. All the procedures operate on single word items, stored as elements 1 to  $n$  of the array  $A$ . The elements are normally so arranged that  $A[i] \leq A[j]$  for  $2 \leq j \leq n$ ,  $i = j \div 2$ . Such an arrange-

Communications of the ACM 347

Một phần "Algorithm 232: Heapsort"

## 2. LỊCH SỬ HÌNH THÀNH CỦA PRIORITY QUEUE

- HeapSort là thuật toán đầu tiên áp dụng cấu trúc: Binary Heap - một dạng cây nhị phân hoàn hảo - để sắp xếp dữ liệu.
- Binary Heap không chỉ hỗ trợ sắp xếp nhanh chóng mà còn trở thành nền tảng cho việc triển khai Priority Queue (hàng đợi ưu tiên).



Complete Binary Tree

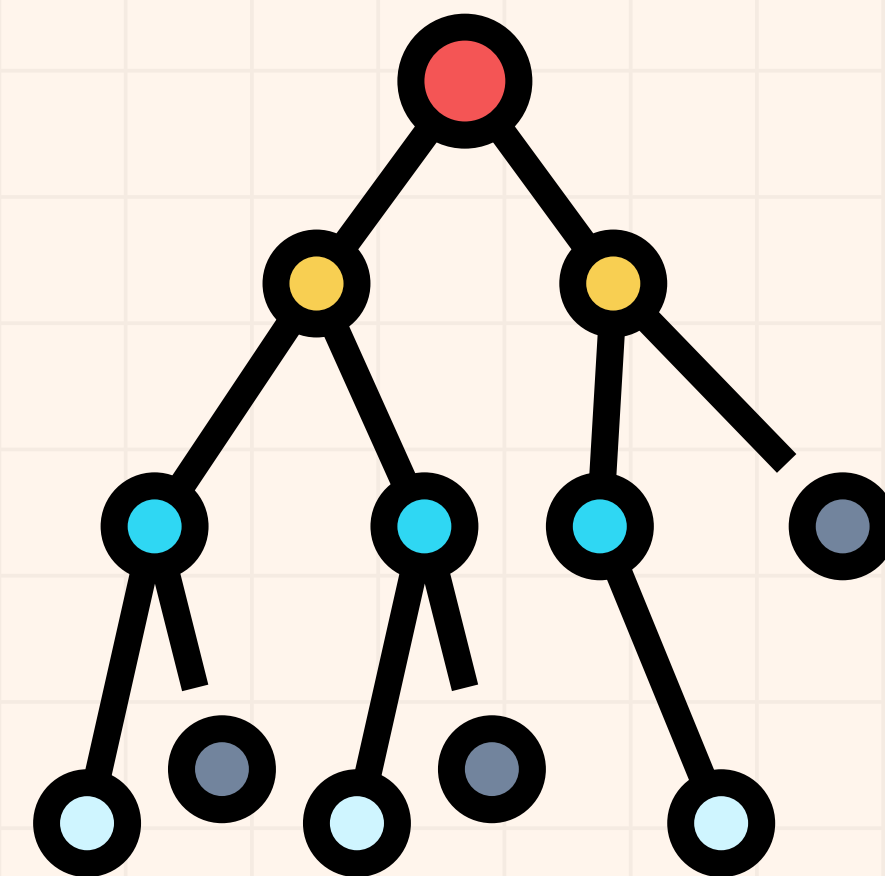


## 2.1 Mối liên hệ Priority Queue và Binary Heap

Cấu trúc binary heap do Williams giới thiệu có hai tính chất nổi bật:

- Truy cập nhanh chóng phần tử quan trọng nhất
- Thao tác hiệu quả

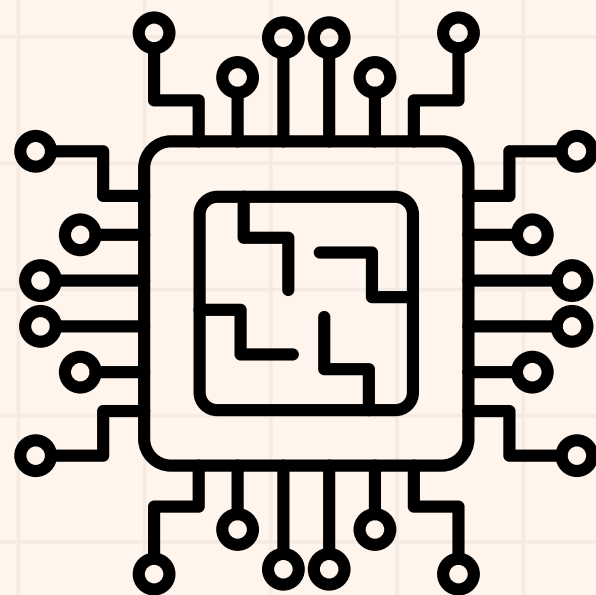
=> Chính những tính chất này đã biến binary heap thành cấu trúc lý tưởng để triển khai hàng đợi ưu tiên



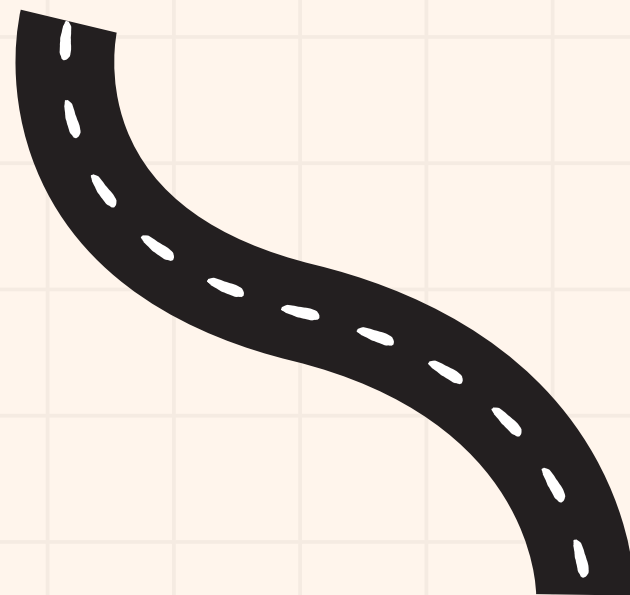
Complete Binary Tree

## 2.2 Ý nghĩa từ công trình nghiên cứu của Williams

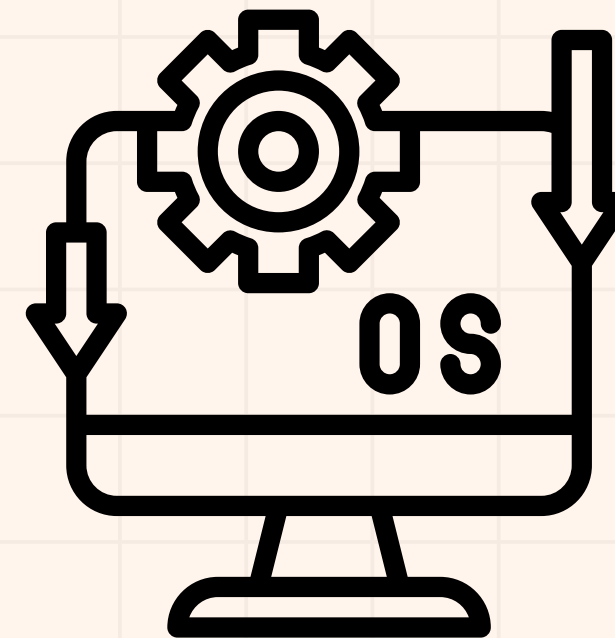
Hệ thống  
lập lịch CPU



Thuật toán tìm  
đường ngắn nhất



Hệ điều hành và  
cơ sở dữ liệu



=> Công trình của Williams không chỉ ảnh hưởng trực tiếp đến việc **tối ưu hóa dữ liệu** mà còn **truyền cảm hứng** cho các nghiên cứu về thuật toán và cấu trúc dữ liệu tiên tiến sau này.

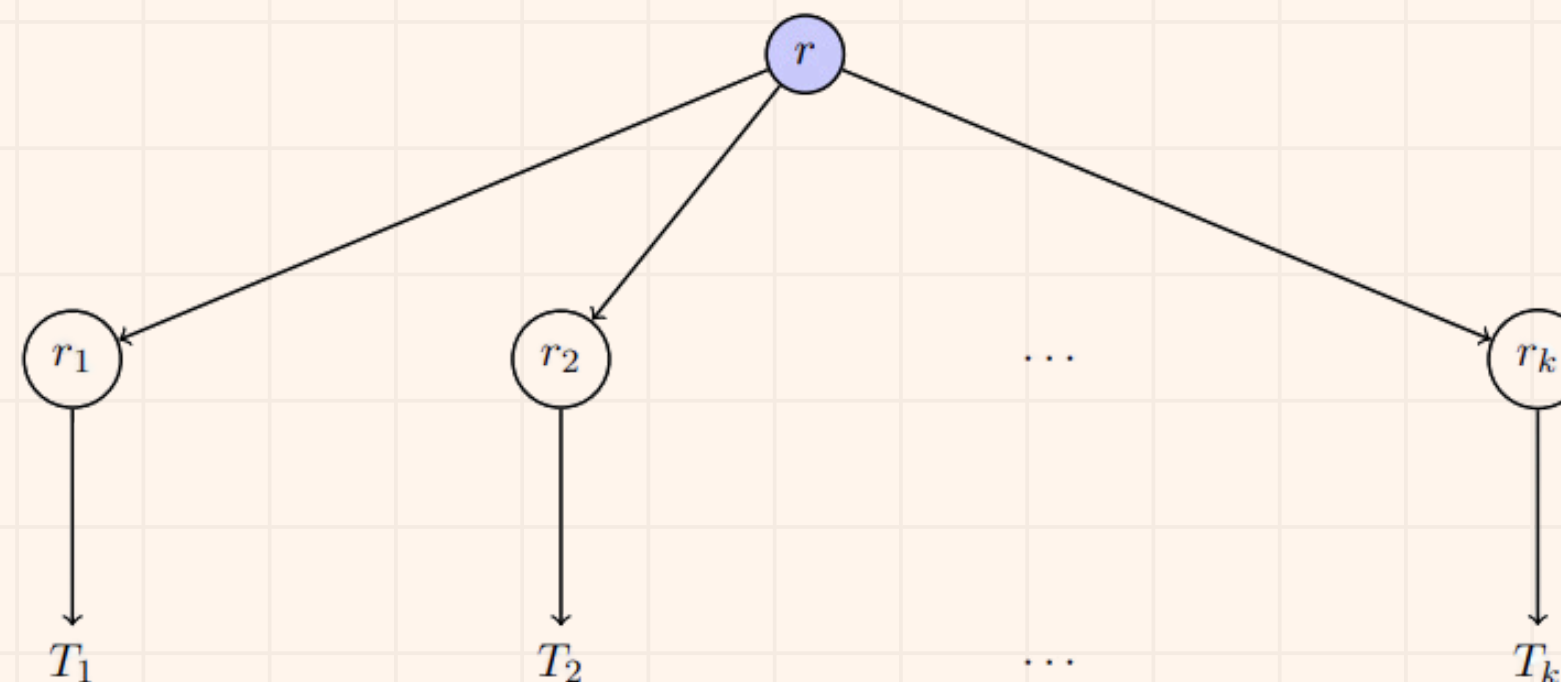


## 3. MỘT SỐ ĐỊNH NGHĨA - KHÁI NIỆM

### 3.1 Cây

#### Định nghĩa:

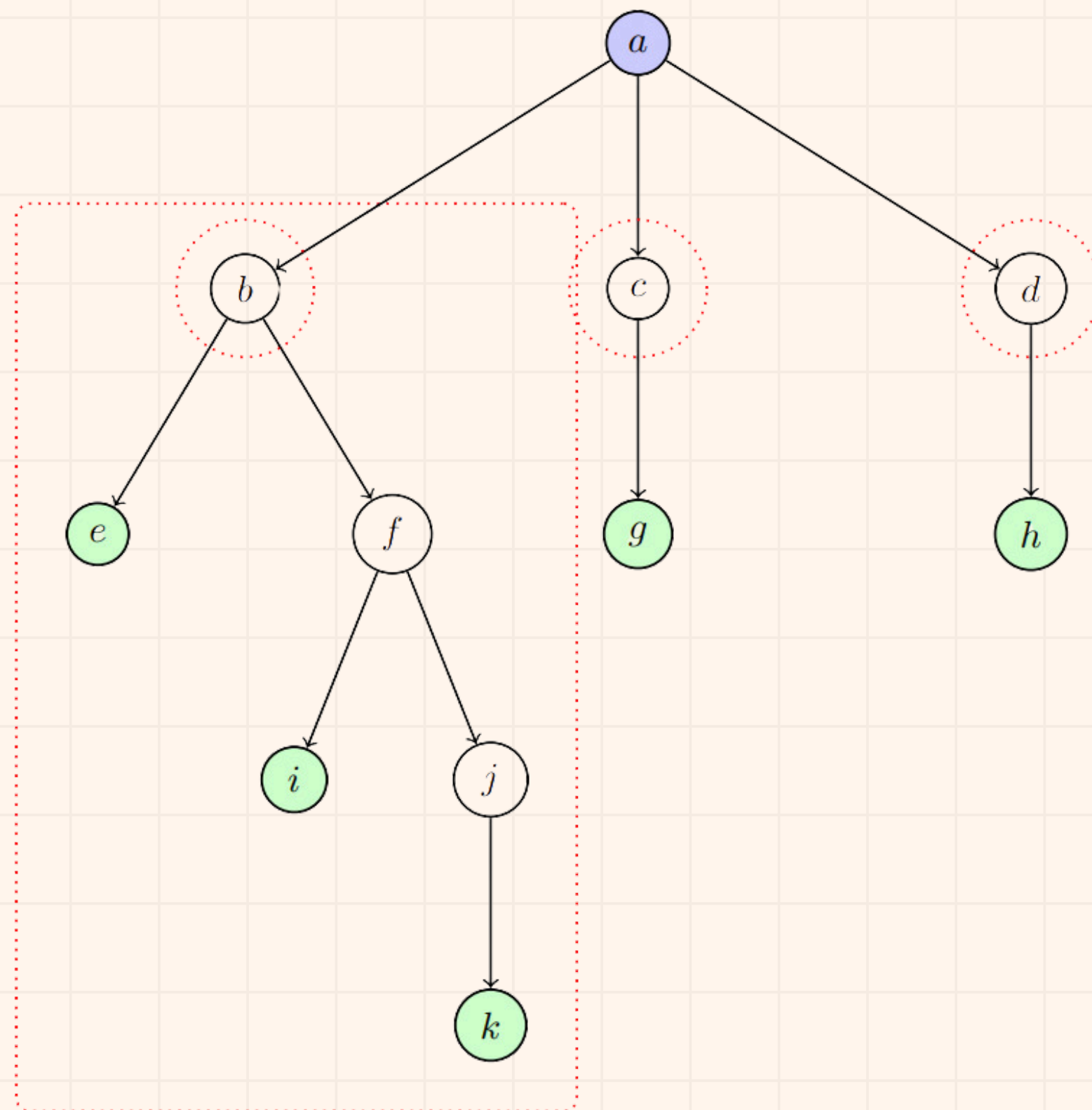
Cây bao gồm các nút, có một nút đặc biệt được gọi là nút gốc (root) và các cạnh nối các nút. Cây được định nghĩa đệ quy như hình bên



*Hình minh họa định nghĩa đệ quy của cây*

## 3. MỘT SỐ ĐỊNH NGHĨA - KHÁI NIỆM

### 3.1 Cây

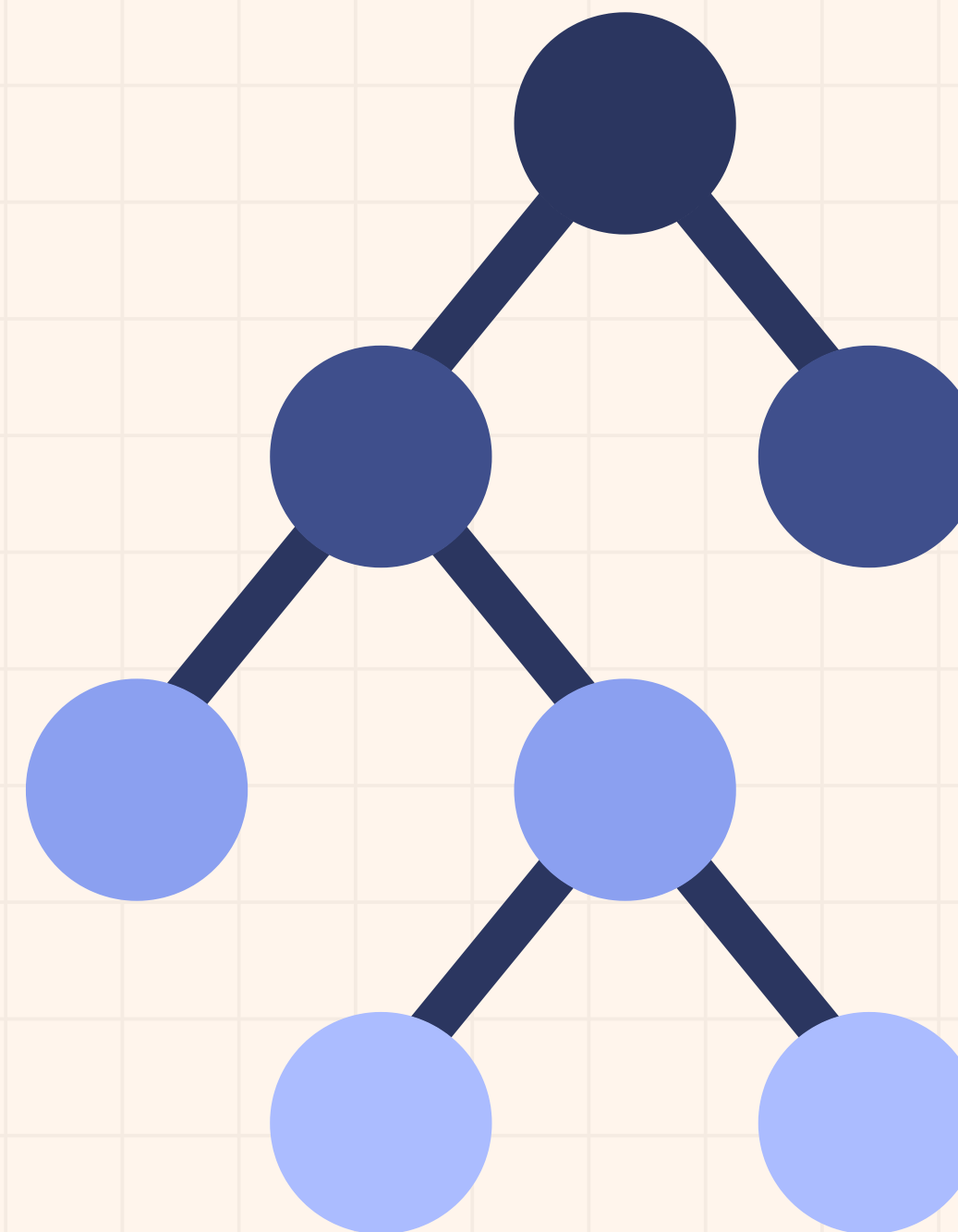


### 3. MỘT SỐ ĐỊNH NGHĨA - KHÁI NIỆM

#### 3.2 Cây nhị phân

**Định nghĩa:**

- Cây nhị phân là một cấu trúc dữ liệu mà mỗi nút cha có tối đa 2 nút con.
- Các thuật ngữ về cây nhị phân giống như các thuật ngữ về cây đã giới thiệu ở trên.



### **3. MỘT SỐ ĐỊNH NGHĨA - KHÁI NIỆM**

#### **3.2 Cây nhị phân**

##### **Các kiểu cây nhị phân:**

- Cây nhị phân đầy đủ (Full binary tree)
- Cây nhị phân hoàn hảo (Perfect binary tree)
- Cây nhị phân hoàn chỉnh (Complete binary tree)
- Cây nhị phân cân bằng (Balanced binary tree)

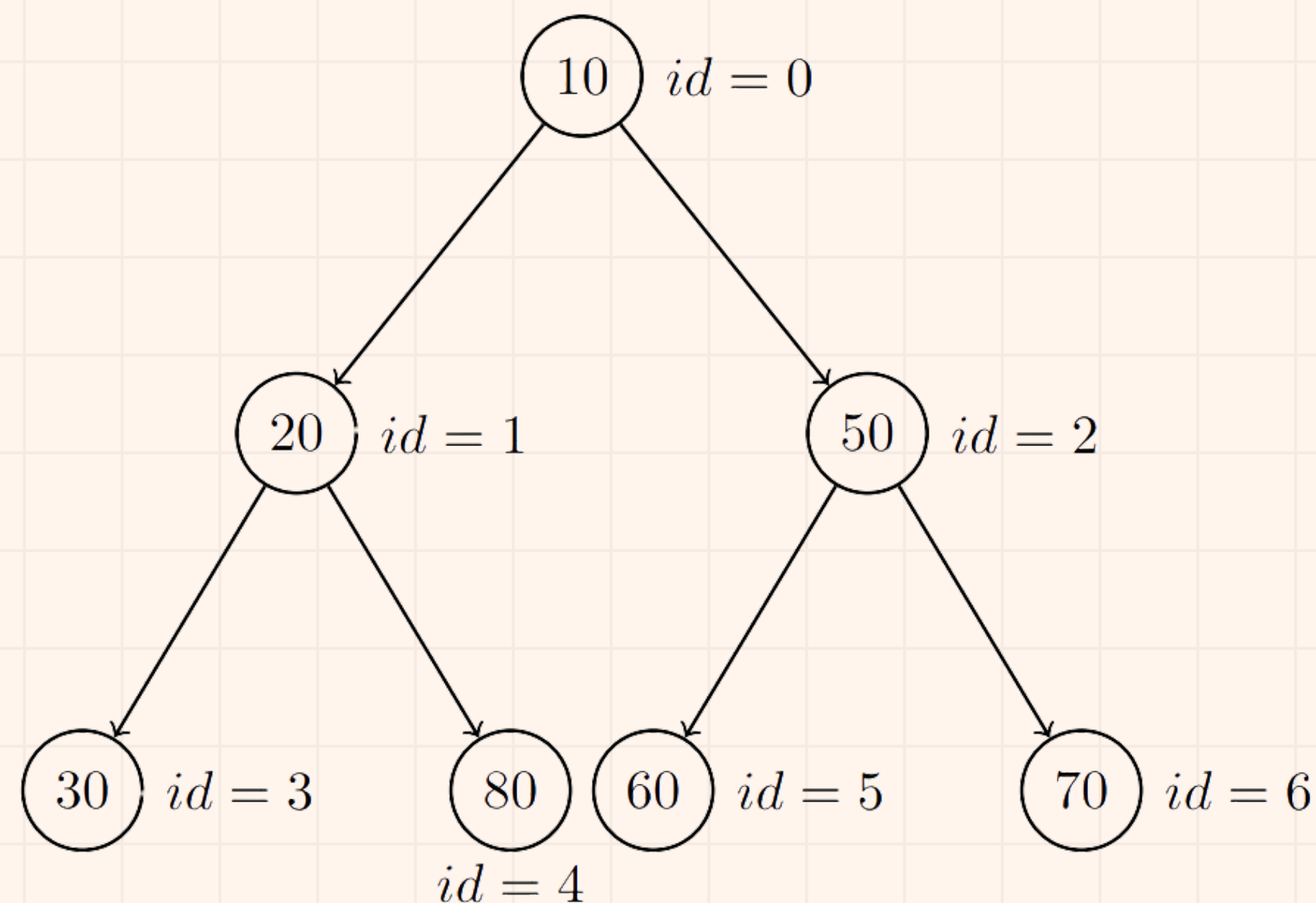
## 3. MỘT SỐ ĐỊNH NGHĨA - KHÁI NIỆM

### 3.3 Binary Heap

#### Định nghĩa

Binary Heap là một cây nhị phân hoàn chỉnh được sử dụng để lưu trữ dữ liệu một cách hiệu quả nhằm lấy được phần tử lớn nhất hoặc nhỏ nhất dựa trên cấu trúc của nó

Binary Heap thường được biểu diễn dưới dạng mảng.



## 3. MỘT SỐ ĐỊNH NGHĨA - KHÁI NIỆM

### 3.3 Binary Heap

#### Phân loại Binary Heap

Max Heap: Cho dãy số  $h_0, h_1, \dots, h_{n-1}$  là một max-heap nếu thỏa mãn:

$$\begin{cases} h_i \geq h_{2i+1} \\ h_i \geq h_{2i+2} \end{cases}, \forall i \in \left[0, \left\lceil \frac{n}{2} \right\rceil - 1\right]$$

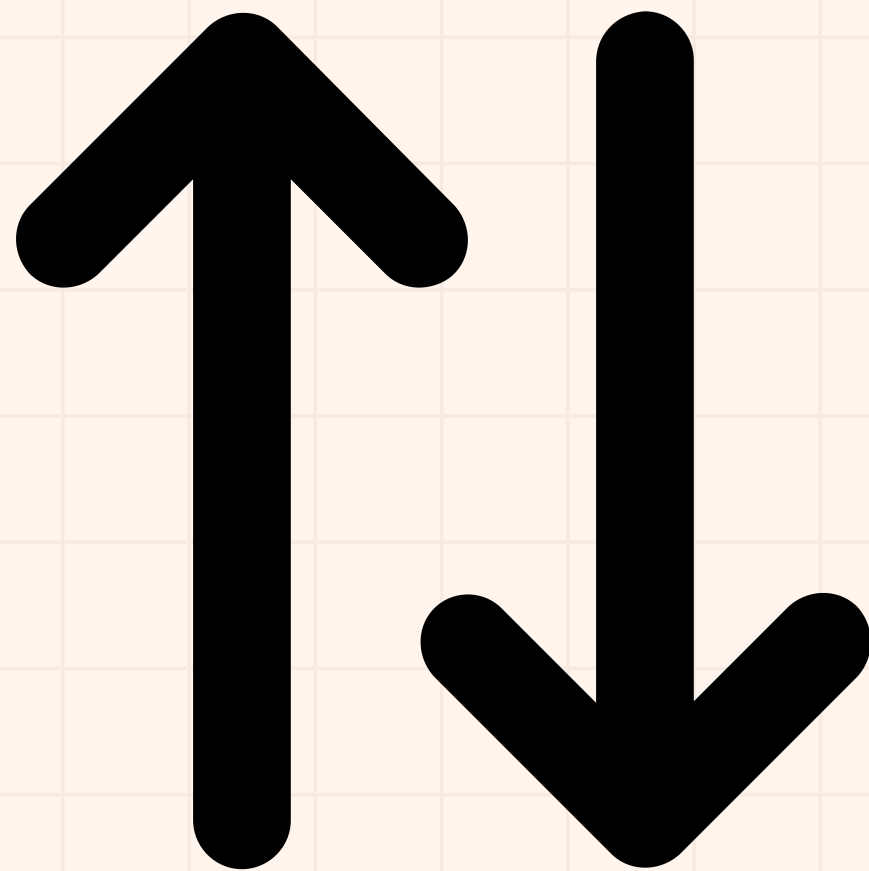
Min Heap: Cho dãy số  $h_0, h_1, \dots, h_{n-1}$  là một min-heap nếu thỏa mãn:

$$\begin{cases} h_i \leq h_{2i+1} \\ h_i \leq h_{2i+2} \end{cases}, \forall i \in \left[0, \left\lceil \frac{n}{2} \right\rceil - 1\right]$$

### 3. MỘT SỐ ĐỊNH NGHĨA - KHÁI NIỆM

#### 3.4 Ứng dụng của Binary Heap

Heap sort



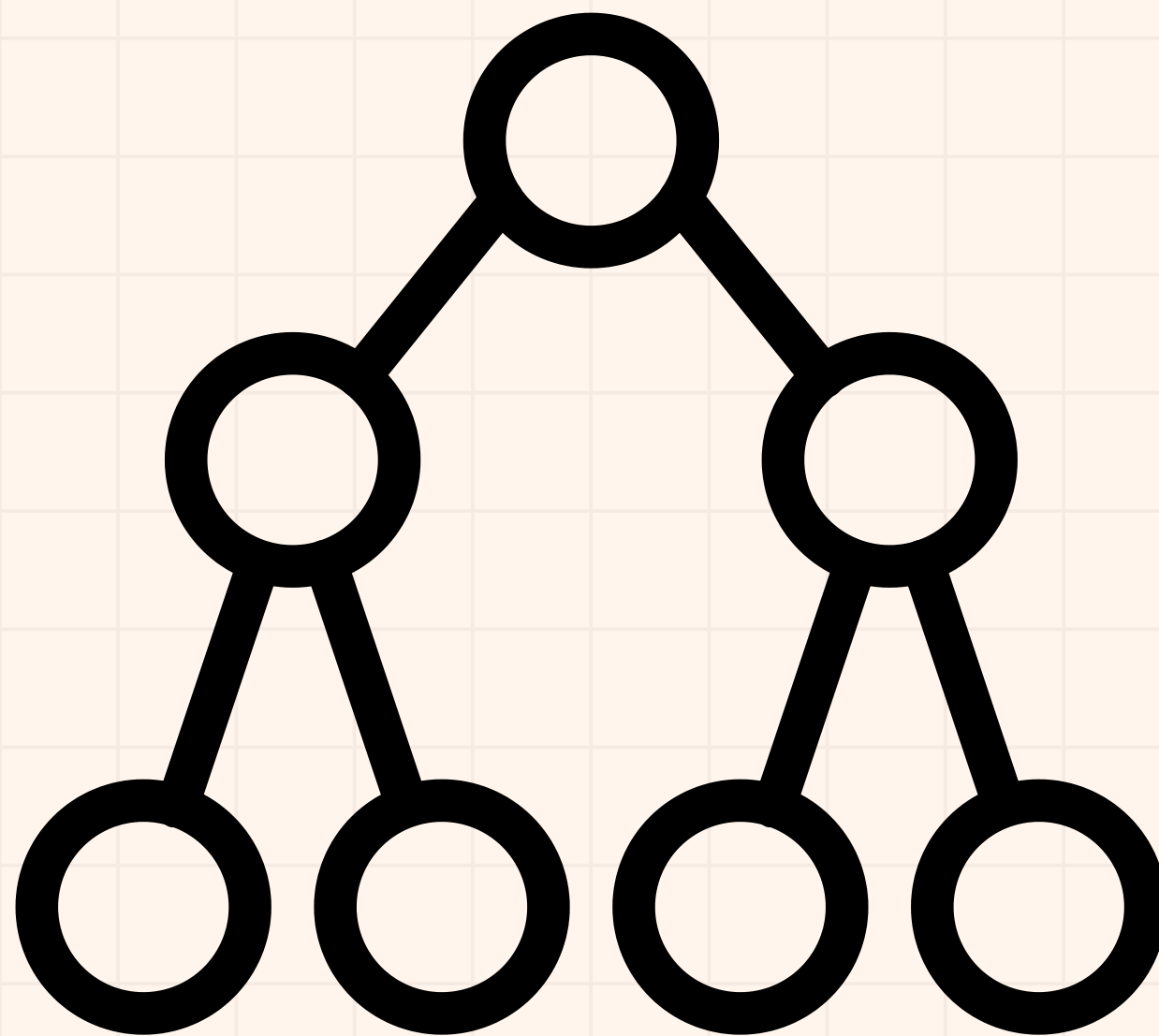
Priority Queue





## 4. PRIORITY QUEUE

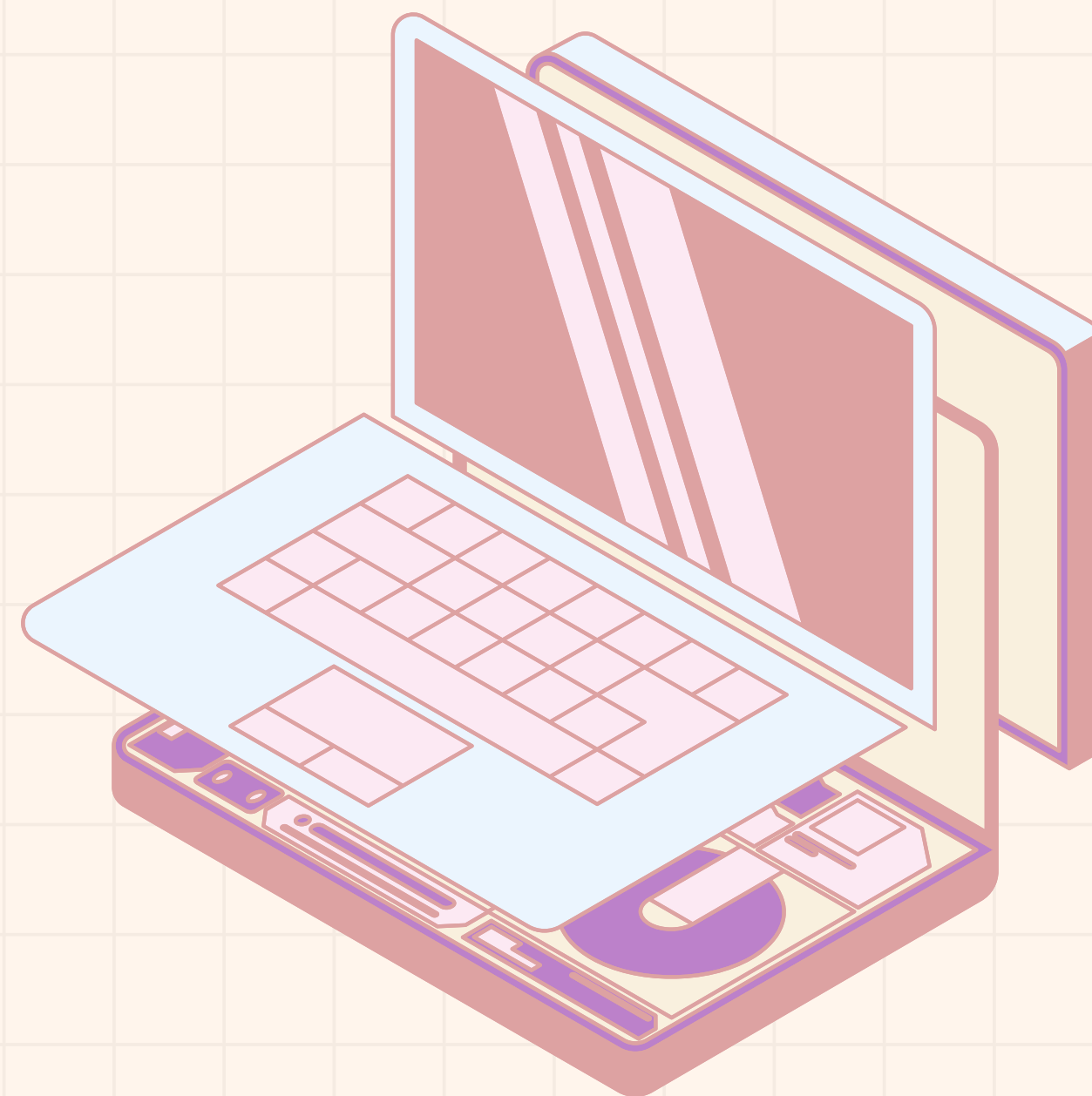
Chúng ta làm việc với hàng đợi ưu tiên có phần tử ở đỉnh hàng đợi là lớn nhất hay chính là một max binary heap.



## 4. PRIORITY QUEUE

Các thao tác chính trong Priority Queue, bao gồm:

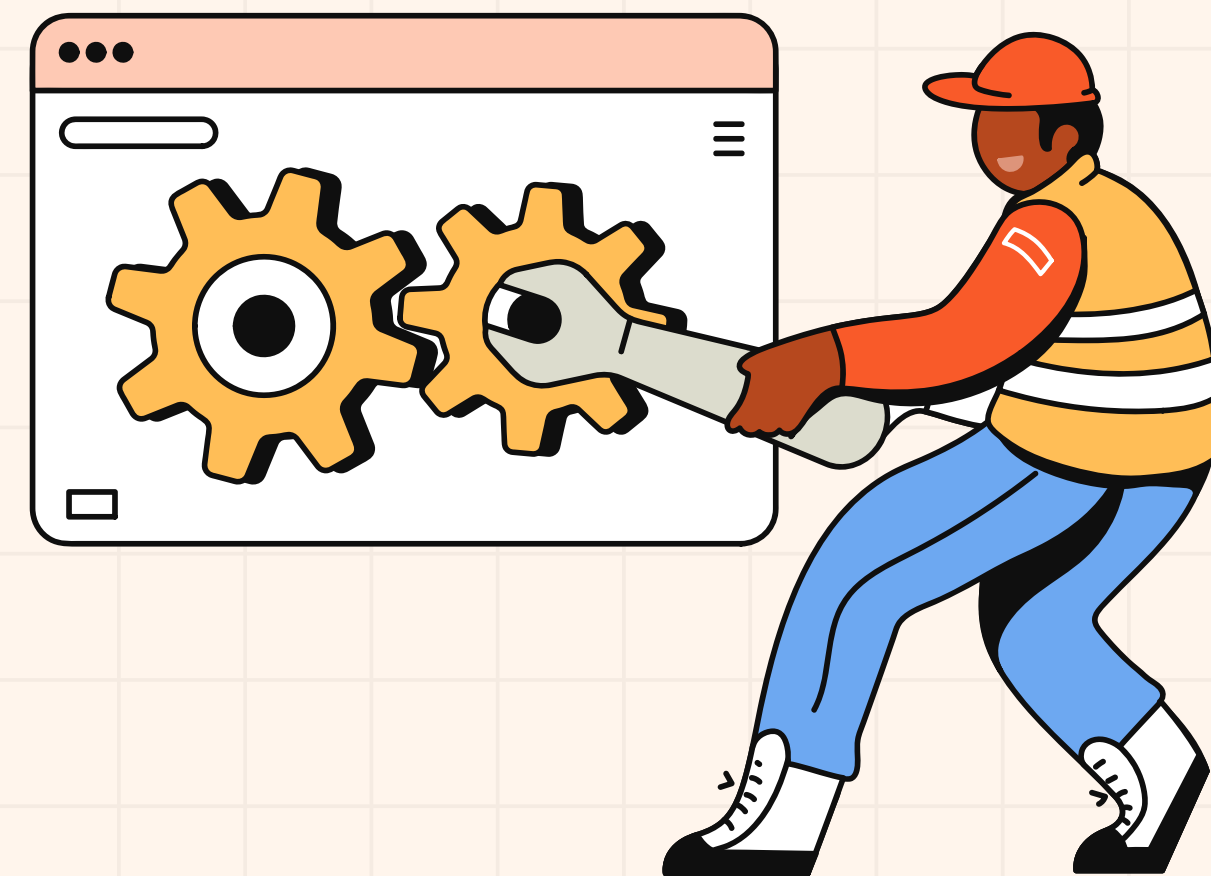
- Push
- Heapify Up
- Pop
- Heapify Down.

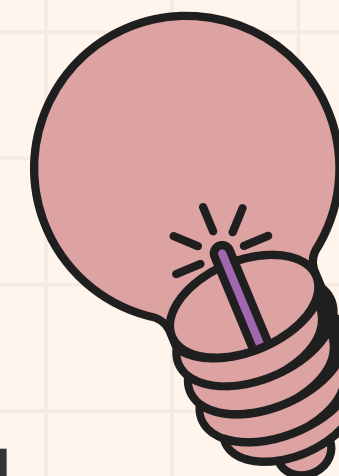
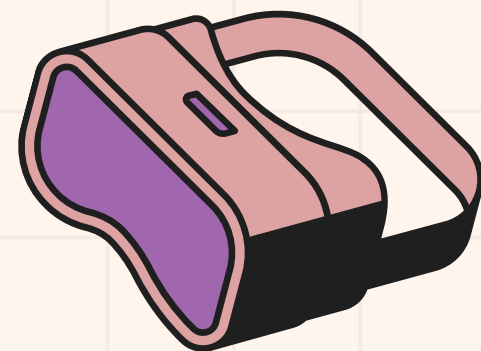


## 4. PRIORITY QUEUE

### Kết luận:

Các thao tác Heapify Up, Pop, và Heapify Down đảm bảo Priority Queue hoạt động đúng, duy trì tính chất của Max - Heap trong mọi trường hợp.





# THANK YOU

