

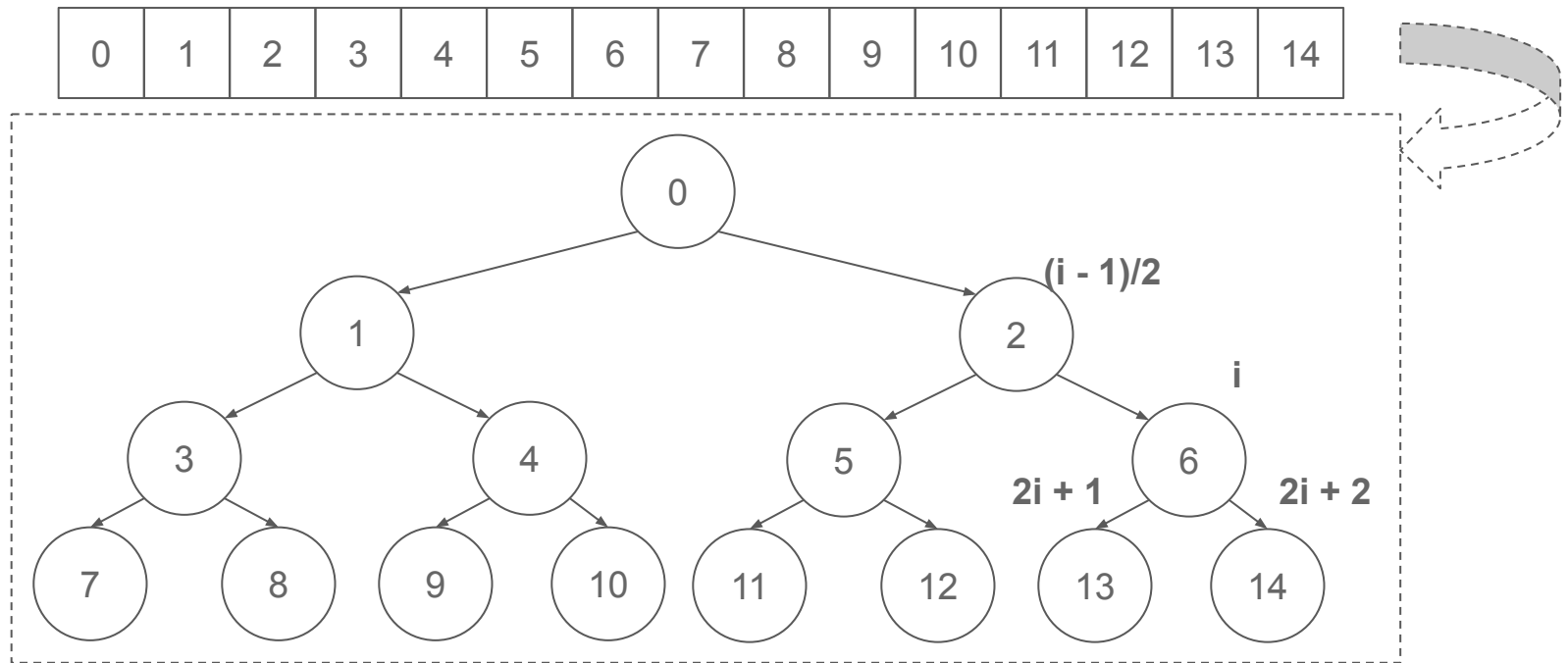
Lập Trình C (cơ bản)

Chương 08. Hàng đợi ưu tiên

Soạn bởi: TS. Nguyễn Bá Ngọc

2021

Mảng và cây nhị phân đầy đủ

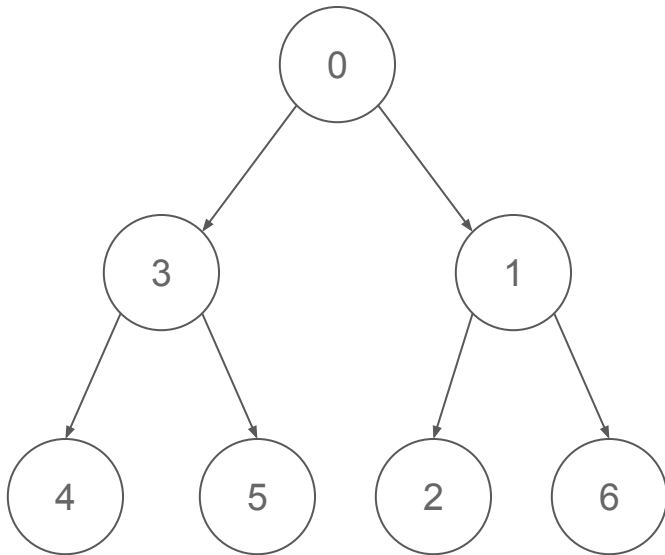


- Có thể coi mảng 1-chiều như cây nhị phân (gần) đầy đủ:
 - Cây nhị phân với tất cả các nút lá có cùng độ cao và được lấp đầy từ trái sang phải, các nút trong có đúng 2 nút con.
 - Nút tương ứng với phần tử có chỉ số i có các liên kết:
 - $\text{Top}(i) = (i - 1)/2$; $\text{Left}(i) = 2 * i + 1$; $\text{Right}(i) = 2 * i + 2$;
 - *(Chỉ số vượt ra ngoài khoảng không có nút tương ứng)*

Cấu trúc Heap

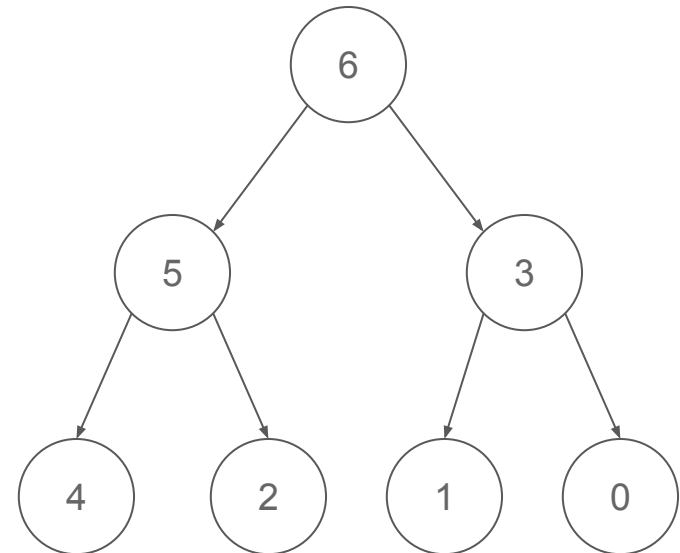
*Heap là 1 dạng mảng cấp phát động, được coi như cây nhị phân đầy đủ, và các nút thỏa mãn **tính chất Heap**:*

- Với tất cả các nút i không phải nút gốc:
 - Heap cực tiểu/Min Heap: $a[\text{Top}(i)] \leq a[i]$ ($\Rightarrow \text{min} = a[0]$).
 - Heap cực đại/Max Heap: $a[\text{Top}(i)] \geq a[i]$ ($\Rightarrow \text{max} = a[0]$).



Heap cực tiểu

0	3	1	4	5	2	6
---	---	---	---	---	---	---



Heap cực đại

6	5	3	4	2	1	0
---	---	---	---	---	---	---

Hàng đợi ưu tiên dựa trên Heap

Khác với hàng đợi FIFO, trong hàng đợi ưu tiên *phần tử có khóa được ưu tiên nhất* là phần tử được lấy ra. Có thể triển khai hàng đợi ưu tiên dựa trên Heap:

- Thao tác enqueue: Thêm phần tử vào sau phần tử cuối cùng trong Heap, sau đó sửa tính chất Heap (nếu cần) bằng cách đẩy phần tử mới về phía nút gốc.
- Thao tác dequeue: Sao lưu nút gốc, sau đó hoán đổi nút gốc với nút cuối cùng trong Heap. Tiếp theo giảm kích thước Heap đi 1 phần tử rồi sửa tính chất Heap từ nút gốc nếu cần. Sau đó trả về nút gốc cũ.
- Thao tác peek: Trả về nút gốc.

Tương ứng với 2 loại Heap, chúng ta cũng có 2 loại hàng đợi ưu tiên: Ưu tiên Min và Ưu tiên Max.

Triển khai hàng đợi ưu tiên trong cgen

- Phân loại hàng đợi ưu tiên

```
typedef enum {  
    PRIORITY_MIN = 0,  
    PRIORITY_MAX,  
    PRIORITY_TYPES_COUNT  
} p1w_types;
```

- `p1w_t p1w_create(p1w_types typ, gtype_cmp_t cmp);`
 - `typ`: Loại hàng đợi ưu tiên, có thể là `PRIORITY_MIN` hoặc `PRIORITY_MAX`.
 - `cmp`: Hàm so sánh 2 giá trị `gtype` theo định dạng `strcmp`.
 - Trả về đối tượng hàng đợi ưu tiên.
- `gtype p1w_peek(p1w_t);`
 - Trả về phần tử sẽ được lấy ra ở lần `dequeue` tiếp theo (là gốc của heap).

Triển khai hàng đợi ưu tiên trong cgen₍₂₎

- `gtype p1w_dequeue(p1w_t);`
 - Trả về và xóa phần tử hiện tại từ hàng đợi.
- `int p1w_enqueue(p1w_t, gtype);`
 - Thêm phần tử vào hàng đợi, trả về 0 nếu thành công hoặc mã lỗi trong trường hợp phát sinh lỗi.
- `void p1w_root(p1w_t, gtype);`
 - Cập nhật phần tử hiện tại (gốc của heap, hiệu quả hơn dequeue sau đó enqueue giá trị mới).
- `long p1w_size(p1w_t);`
 - Trả về số lượng phần tử trong hàng đợi.
- Các macro:
 - `p1w_is_empty(q)` tương đương với `(p1w_size(q) == 0)`
 - `p1w_free(q)` giải phóng bộ nhớ động đã cấp phát cho q.

Bài tập 8.1. Lấy top K lớn nhất

Cho tệp văn bản inp.txt chứa n số thực, với mỗi giá trị được ngăn cách bởi 1 khoảng trắng.

$$10 \leq n \leq 10^7$$

Yêu cầu: Chương trình hỏi người dùng nhập vào số K, sau đó:

- 1) Đưa ra K giá trị lớn nhất trong n giá trị có trong inp.txt.
- 2) Đo thời gian xử lý.

Bài tập 8.2. Lấy top K lớn nhất

Cho tệp văn bản inp.txt chứa n số thực, với mỗi giá trị được ngăn cách bởi 1 khoảng trắng.

$$10 \leq n \leq 10^7$$

Yêu cầu: Chương trình hỏi người dùng nhập vào số K, sau đó so sánh thời gian tìm K giá trị lớn nhất theo 2 cách.

- 1) Sắp xếp rồi lấy top K
- 2) Sử dụng Heap/hàng đợi ưu tiên với giới hạn K phần tử.

