Toàn rời rạc (CE23101)

Giải thuật trên cây

Giải thuật di chuyển trên cây

Có ba phép duyệt để di chuyển trên cây có gốc.

Giả sử ta đang ở một nút r với các cây con là $\mathcal{T}_1^{(r)}, \mathcal{T}_2^{(r)}, \dots, \mathcal{T}_m^{(r)}$ theo thứ tự từ trái sang phải.

- Duyệt trước (pre-order). Duyệt r trước, sau đó duyệt $\mathcal{T}_k^{(r)}$ theo thứ tự $k = 1, 2, \dots, m$.
- Duyệt giữa (in-order). Duyệt $\mathcal{T}_1^{(r)}$ trước, sau đó duyệt r, sau đó duyệt $\mathcal{T}_k^{(r)}$ theo thứ tự $k = 2, 3, \dots, m$.
- Duyệt cuối (post-order). Duyệt $\mathcal{T}_k^{(r)}$ theo thứ tự $k = 1, 2, \dots, m$, sau đó duyệt r.

2

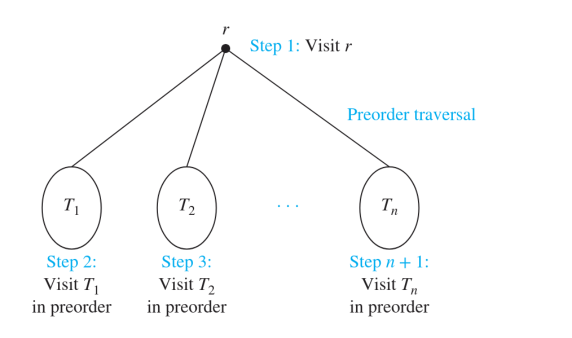


Figure 1: Duyệt trước (pre-order). Hình từ sách của Ken Rosen

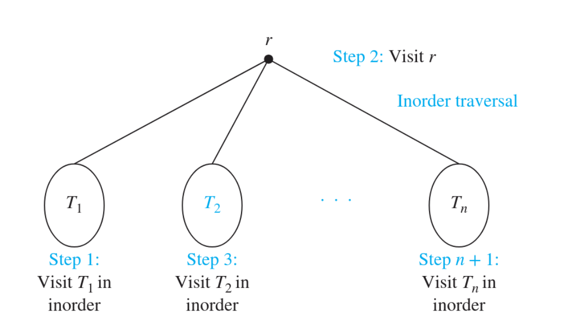


Figure 2: Duyệt giữa (pre-order). Hình từ sách của Ken Rosen

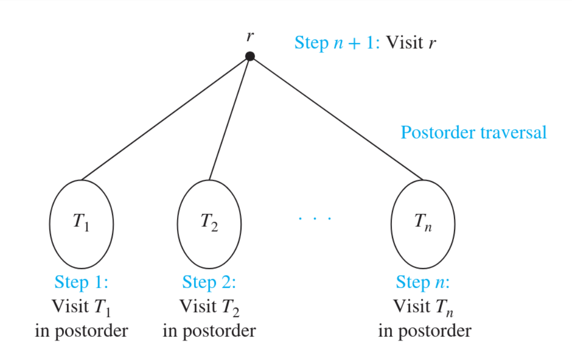


Figure 3: Duyệt cuối (post-order). Hình từ sách của Ken Rosen

Ví dụ 1:

Cho $\mathcal{T}$ như sau. Viết thứ tự các đỉnh của $\mathcal{T}$ theo phương pháp

- duyệt trước (pre-order)
- duyệt giữa (in-order)
- duyệt sau (post-order)

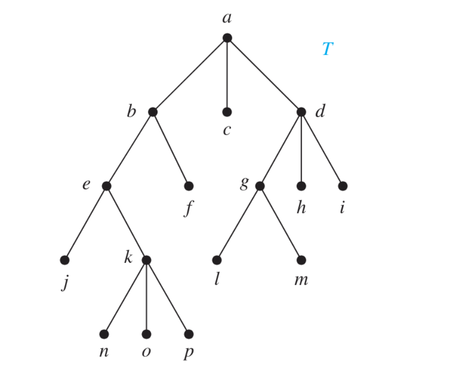


Figure 4: Một cây có gốc. Hình từ sách của Ken Rosen

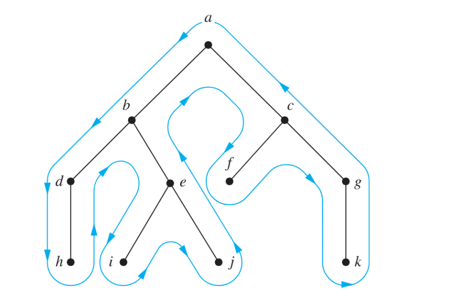


Figure 5: Mẹo để nhớ cách duyệt. Hình từ sách của Ken Rosen

Cho cây tìm kiếm nhị phân sau đây. Tìm các giá trị của $\mathcal{T}$ theo phương pháp duyệt giữa.

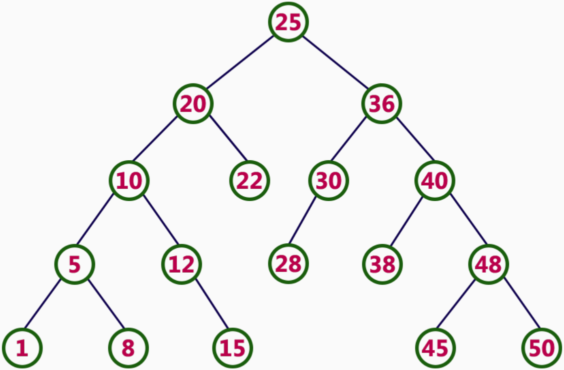


Figure 6: Cây tìm kiếm nhị phân (binary search tree)

Ký pháp (nghịch đảo) Ba Lan

Vẽ cây các phép tính khi tính giá trị của $(7-(2\times 3))^4+(9/3)$.

Ký pháp (nghịch đảo) Ba Lan

Vẽ cây các phép tính khi tính giá trị của $(7 - (2 \times 3))^4 + (9/3)$.

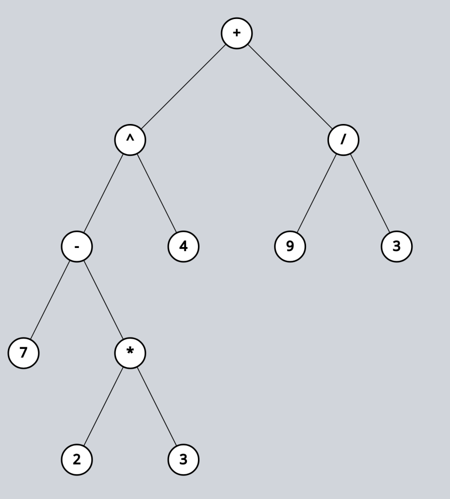
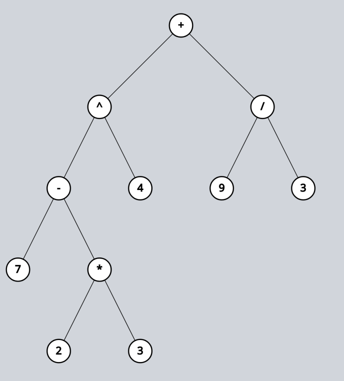


Figure 7: Cây tính toán giá trị $(7 - (2 \times 3))^4 + (9/3)$.



Từ cây trên, dùng phép duyệt

- pre-order chúng ta có ký pháp Ba Lan

$$+ \uparrow - 7 * 2\,3\,4 / 9\,3$$

- post-order chúng ta có ký pháp nghịch đảo Ba Lan

$$7\,2\,3 * - 4 \uparrow 9\,3 / +$$

10

Sắp xếp tô-pô

Giả sử ta có một đồ thị $G = (V, E)$ có hướng và không có chu trình (directed acyclic graph hoặc DAG) và ta muốn sắp xếp các đỉnh sao cho nếu $(u, v) \in E$ thì u xuất hiện trước v.

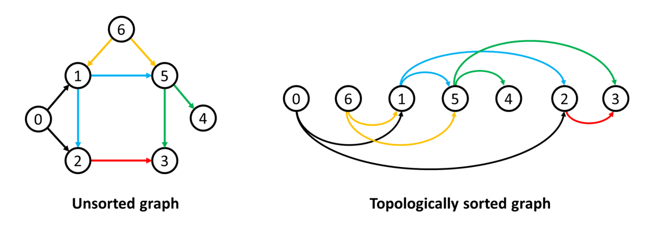


Figure 8: Hình từ trang web của thư viện VNOI.

Sắp xếp tô-pô được dùng khi lên kế hoạch các việc phải làm.

Ta có thể giải bài toán sắp xếp tô-pô bằng phương pháp duyệt sau. Ta chọn gốc là bất kỳ một đỉnh v nào mà $\deg_{\text{in}}(v) = 0$.

```
L ← danh sách rỗng (cuối cùng sẽ chứa thứ tự sắp xếp)
S ← tập hợp các nút không có cung vào
for each nút n trong S do
    thăm(n)
function thăm(nút n)
    if chưa thăm n then
        đánh dấu n là đã thăm
        for each nút m sao cho có cung từ n đến m do
            thăm(m)
        chèn n vào L
```

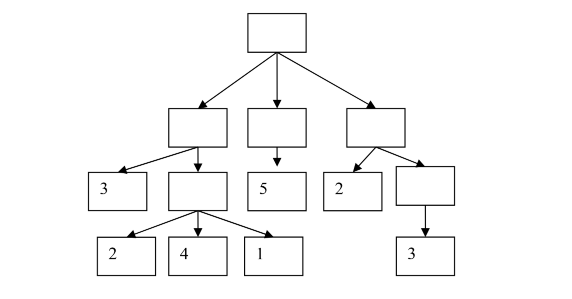
Figure 9: Hình từ Wikipedia

12

Cây có trọng số

Một cây có trọng số là một cây có gốc mà các lá ℓ được gán một trọng số $w(\ell)$ không âm. Trọng số của cây được định nghĩa là

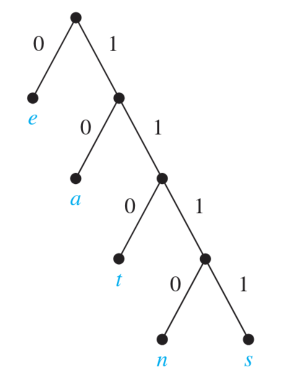
$$w(\mathcal{T}) = \sum_{\ell \text{ là lá}} w(\ell) \times \mathrm{level}(\ell).$$



13

Giả sử ta có một tập hợp các giá trị $w_1, w_2, \ldots, w_n$ không âm, làm sao để tạo một cây nhị phân $\mathcal{T}$ mà các lá tương ứng với $\{w_i\}$ và $w(\mathcal{T})$ là nhỏ nhất ?

Một ứng dụng của bài toán trên là để mã hõa các ký tự $a, b, c, \ldots$ bằng các chuỗi nhị phân kiểu $a = 0, b = 10, c = 11, \ldots$ sao cho (1) có thể luôn luôn giải mã và (2) chiều dài trung bình của một mã ngẫu nhiên là ngắn nhất.



Giải thuật Huffman

- Khởi tạo: Tạo một rừng gồm n cây, mỗi cây chỉ có một nút gốc, mỗi nút gốc tương ứng với một ký tự và có trọng số là tần số/tần suất của ký tự đó $W(i)$.
- Lặp:
  - Mỗi bước sau thực hiện cho đến khi rừng chỉ còn một cây:
  - Chọn hai cây có trọng số ở gốc nhỏ nhất hợp thành một cây bằng cách thêm một gốc mới nối với hai gốc đã chọn. Trọng số của gốc mới bằng tổng trọng số của hai gốc tạo thành nó.

Figure 10: Hình từ trang Wikipedia

15

Ví dụ 2:

Tìm mã Huffman cho các ký tự a, b, c, d, e với trọng số 0.1, 0.15, 0.3, 0.16, 0.29.

16

Ví dụ 3:

Tìm mã Huffman cho các ký tự a, b, c, d, e với trọng số 0.1, 0.15, 0.3, 0.16, 0.29.

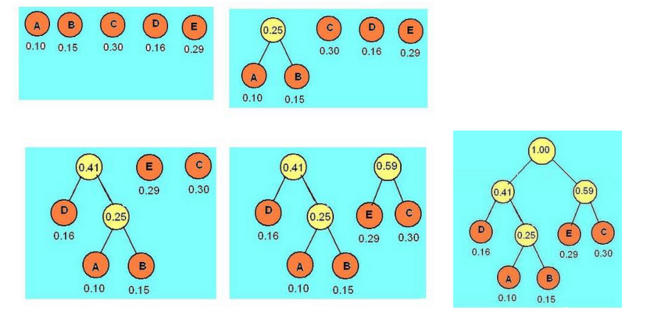


Figure 11: Hình từ trang Wikipedia

Chứng minh sự tối ưu của giải thuật Huffman

Gọi $S = \{w_1, w_2, \ldots, w_n\}$.

Ta bắt đầu bằng hai mệnh đề như sau

Mệnh đề

Cho $\mathcal{T}$ là một cây nhị phân, và x, y là hai đỉnh lá. Gọi $\mathcal{T}'$ là cây mà ta có khi hoán đổi vị trí của lá x và y. Ta sẽ có

$$w(\mathcal{T}') - w(\mathcal{T}) = (w_y - w_x)(\text{level}(x) - \text{level}(y))$$

$$\begin{aligned} w(\mathcal{T}') - w(\mathcal{T}) &= w_x \text{level}(y) + w_y \text{level}(x) - w_x \text{level}(x) - \mathrm{w}_y \text{level}(y) \\ &= (w_y - w_x)(\text{level}(x) - \text{level}(y)) \end{aligned}$$

17

Mệnh đề

Tồn tại một cây nhị phân tối ưu sao cho nếu w_x, w_y là hai giá trị nhỏ nhất của S thì x và y là anh em.

Gọi $\mathcal{T}$ là một cây tối ưu và w_x, w_y là hai giá trị nhỏ nhất của S. Nếu x, y là anh em thì không còn gì để chứng minh.

Giả sử (không mất tính tổng quát) là $\text{level}(x) \geq \text{level}(y)$.

Ta xét hai trường hợp

1. Nếu x có anh em là $z \neq y$. Gọi $\mathcal{T}'$ là cây khi ta hoán đổi vị trí của z và y. Ta có

$$w(\mathcal{T}') - w(\mathcal{T}) = (w_z - w_y)(\text{level}(y) - \text{level}(x)) \leq 0.$$

Suy ra $\mathcal{T}'$ cũng phải tối ưu. Hơn nưa x, y là anh em trong $\mathcal{T}'$.

2. Nếu x không có anh em thì khi ta giảm level của x sẽ dẫn đến mâu thuẫn của việc $\mathcal{T}$ tối ưu.

Ta sẽ sử dụng nguyên tắc quy nạp.

Bước cơ sở: Chỉ có w_1 và w_2 và giải thuật Huffman đưa ra một cây nhị phân với hai lá và độ cao của các lá bằng nhau.

Suy ra cây đó phải tối ưu.

Mệnh đề quy nạp: Giả sử giải thuật Huffman tối ưu cho mọi $\{w_1, w_2, \ldots, w_k\}$ với $k \leq n$.

Bước quy nạp: Cho một $S = \{w_1, w_2, \ldots, w_{n+1}\}$. Gọi x và y là hai phần tử với w_x, w_y là hai giá trị nhỏ nhất trong S.

Bước đầu tiên của giải thuật Huffman ta sẽ nhập x và y thành một ký hiệu là z.

Đặt $S' = S \setminus \{w_x, w_y\} \cup \{w_z\}$.

Gọi $\mathcal{T}$ là cây nhị phân cho S và $\mathcal{T}'$ là cây nhị phân cho S' (khi dùng giải thuật Huffman).

Chú ý là $\mathcal{T}'$ là tối ưu cho S' (mệnh đề quy nạp)

Ta có $w(\mathcal{T}) = w(\mathcal{T}') + w_x + w_y$.

Nếu $\mathcal{T}$ không phải là tối ưu thì có một cây $\tilde{\mathcal{T}}$ với $w(\mathcal{T}) > w(\tilde{\mathcal{T}})$.

Từ mệnh đề trước ta có thể giả sử x và y là anh em trong $\tilde{\mathcal{T}}$.

Tạo $\tilde{\mathcal{T}}'$ bằng cách gộp x với y. $\tilde{\mathcal{T}}'$ là một cây nhị phân cho S'.

Ta sẽ có

$$w(\tilde{\mathcal{T}}') = w(\tilde{\mathcal{T}}) - w_x - w_y < w(\mathcal{T}) - w_x - w_y = w(\mathcal{T}')$$

và đây là mâu thuẫn lại với giả thiết quy nạp.

Suy ra $\mathcal{T}$ phải là tối ưu.

20

Cây bào trùm nhỏ nhất

Giả sử ta có một đồ thị $G = (V, E)$ với hàm trọng số $w\colon E \mapsto \mathbb{R}_{\geq 0}$ cho các cạnh thuộc E. Ta muốn tìm một cây bao trùm $\mathcal{T}$ sao cho $\sum_{e \in \mathcal{T}} w(e)$ là nhỏ nhất.

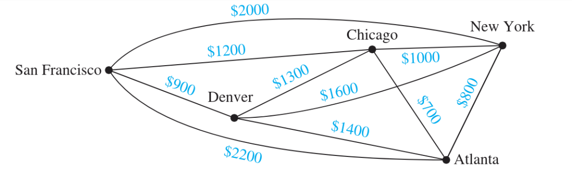
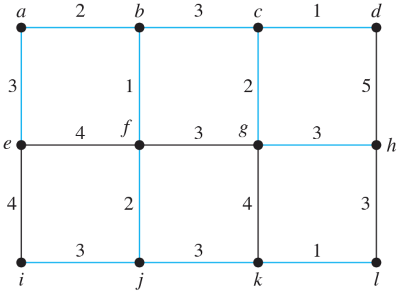


Figure 12: Hình từ sách toán rời rạc của Ken Rosen

Bài toán cây bào trùm nhỏ nhất có thể giải được bằng giải thuật Kruskal hay giải thuật Prim.

- Khởi tạo rừng F (tập hợp các cây), trong đó mỗi đỉnh của G tạo thành một cây riêng biệt.
- Khởi tạo tập S chứa tất cả các cạnh của G.
- Chừng nào S còn **khác rỗng** và F gồm **hơn một cây**
  - Xóa cạnh nhỏ nhất trong S
  - Nếu cạnh đó nối hai cây khác nhau trong F, thì thêm nó vào F và hợp hai cây kề với nó làm một
  - Nếu không thì loại bỏ cạnh đó.

Figure 13: Giải thuật Kruskal. Hình từ trang web của thư viện VNOI.



(a)

Choice	Edge	Weight
1	$\{c, d\}$	1
2	$\{k, l\}$	1
3	$\{b, f\}$	1
4	$\{c, g\}$	2
5	$\{a, b\}$	2
6	$\{f, j\}$	2
7	$\{b, c\}$	3
8	$\{j, k\}$	3
9	$\{g, h\}$	3
10	$\{i, j\}$	3
11	$\{a, e\}$	3
	Total:	24

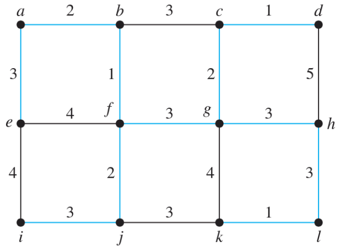
(b)

Figure 14: Hình từ sách toán rời rạc của Ken Rosen

- Dữ liệu vào: Một đồ thị có trọng số liên thông với tập hợp đỉnh *V* và tập hợp cạnh *E* (trọng số có thể âm). Đồng thời cũng dùng *V* và *E* để ký hiệu số đỉnh và số cạnh của đồ thị.
- Khởi tạo: $V_{\text{mới}} = \{x\}$, trong đó *x* là một đỉnh bất kì (đỉnh bắt đầu) trong *V*, $E_{\text{mới}} = \{\}$
- Lặp lại cho tới khi $V_{\text{mới}} = V$:
  - Chọn cạnh (*u*, *v*) có trọng số nhỏ nhất thỏa mãn *u* thuộc $V_{\text{mới}}$ và *v* không thuộc $V_{\text{mới}}$ (nếu có nhiều cạnh như vậy thì chọn một cạnh bất kì trong chúng)
  - Thêm *v* vào $V_{\text{mới}}$, và thêm cạnh (*u*, *v*) vào $E_{\text{mới}}$
- Dữ liệu ra: $V_{\text{mới}}$ và $E_{\text{mới}}$ là tập hợp đỉnh và tập hợp cạnh của một cây bao trùm nhỏ nhất

Figure 15: Giải thuật Prim. Hình từ trang Wikipedia

24



(a)

Choice	Edge	Weight
1	$\{b, f\}$	1
2	$\{a, b\}$	2
3	$\{f, j\}$	2
4	$\{a, e\}$	3
5	$\{i, j\}$	3
6	$\{f, g\}$	3
7	$\{c, g\}$	2
8	$\{c, d\}$	1
9	$\{g, h\}$	3
10	$\{h, l\}$	3
11	$\{k, l\}$	1
	Total:	24

(b)

Figure 16: Hình từ sách toán rời rạc của Ken Rosen

25