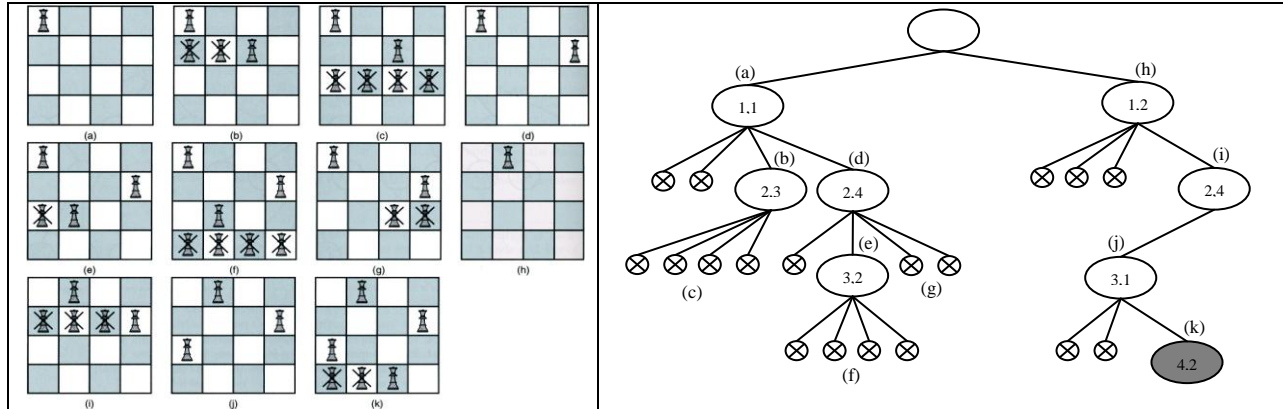


Chương 4: Kỹ thuật Quay lui

4.1. Bài toán n -Hậu



Giải thuật

```

n_Queens(i) {
    for (j = 1; j <= n; j++)
        if (a[j] && b[i + j] && c[i - j]) {
            x[i] = j;
            a[j] = b[i + j] = c[i - j] = false;
            if (i == n)
                Print(x);
            else
                n_Queens(i + 1);
            a[j] = b[i + j] = c[i - j] = true;
        }
}
n_Queens(1);

```

4.2. Bài toán Ngựa đi tuần

	4		3		-2
5				2	-1
		1			0
6				1	1
	7		0		2
-2	-1	0	1	2	

Giải thuật

```

KnightTour(int i, int r, int c) {
    for (k = 1; k <= 8; k++) {
        u = r + dong[k];
        v = c + cot[k];
    }
}

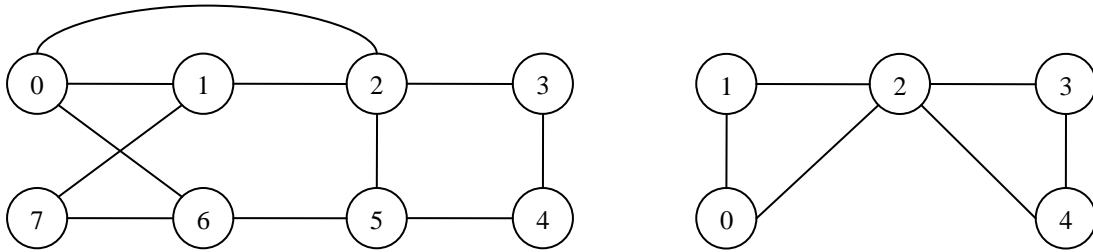
```

```

    if ((1 ≤ u, v ≤ n) && (h[u][v] == 0)) {
        h[u][v] = i;
        if (i == n2)
            Print(h);
        else
            KnightTour(i + 1, u, v);
        h[u][v] = 0;
    }
}
h[r0][c0] = 1;
KnightTour(2, r0, c0);

```

4.3. Chu trình Hamilton



Giải thuật

```

bool promising(int pos, int v) {
    if (pos == n - 1 && G[v][path[0]] == false)
        return false;
    else
        if (G[path[pos - 1]][v] == false)
            return false;
        else
            for (int i = 0; i < pos; i++)
                if (path[i] == v)
                    return false;
    return true;
}

Hamiltonian(bool G[0 .. n-1][ 0 .. n-1], int path[0 .. n-1], int pos) {
    if (pos == n)
        print(path);
    else
        for (int v = 0; v < n; v++)
            if (promising(pos, v)) {
                path[pos] = v;
                Hamiltonian(G, path, pos + 1);
            }
}

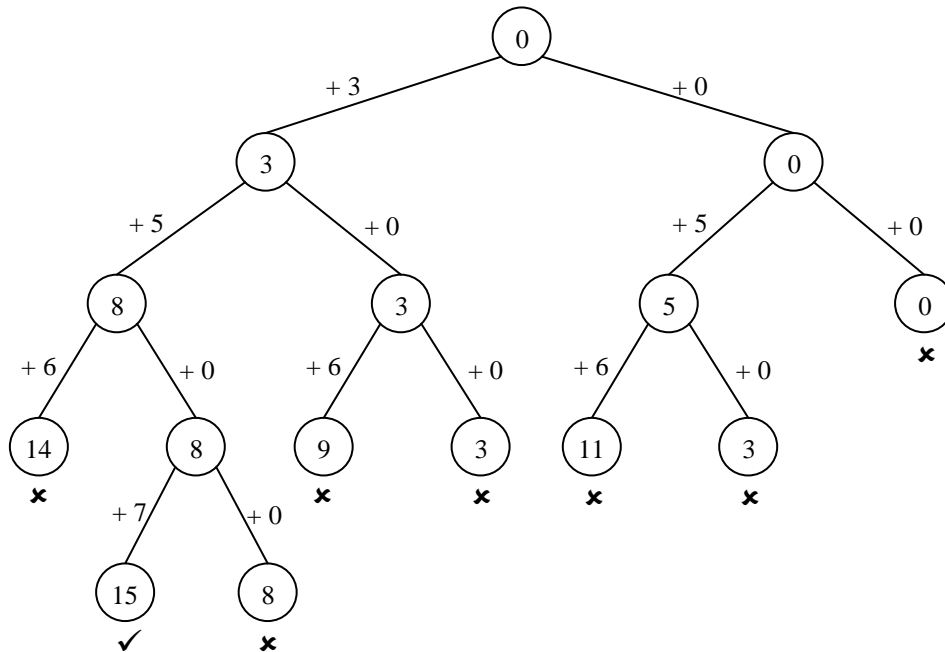
path[0] = 0;
Hamiltonian(G, path, 1);

```

4.4. Bài toán Tổng các tập con

Dạng 1

Ví dụ: $\mathcal{W} = \{3, 5, 6, 7\}$ và $w = 15$.



Giải thuật

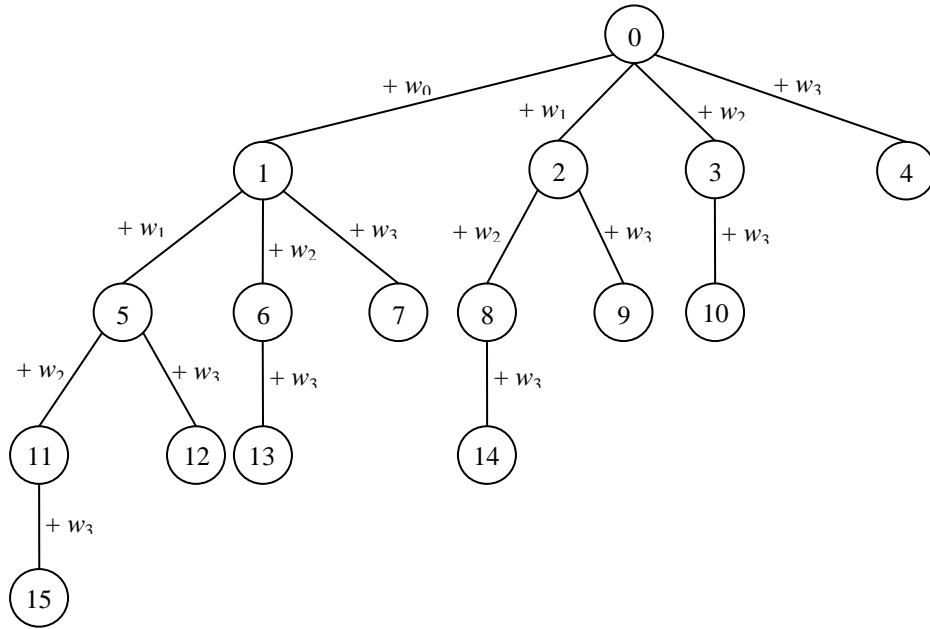
```
sum_of_subsets(int k, int sum, int total, bool include[]) {
    if (sum == W)
        print(include);
    else
        if ((sum + total >= W) && (sum + Value[k] <= W)) {
            include[k] = true;
            sum_of_subsets(k + 1, sum + Value[k], total - Value[k], include);

            include[k] = false;
            sum_of_subsets(k + 1, sum, total - Value[k], include);
        }
}

bool include[0 .. n-1] = {false};
int total =  $\sum_{i=0}^{n-1} \text{Value}[i]$ ;
sum_of_subsets(0, 0, total, include);
```

Dạng 2

Ví dụ: $\mathcal{W} = \{w_0, w_1, w_2, w_3\}$.



Giải thuật (Quay lui vết cạn)

```

subset_sum(include[0 .. n-1], size, sum, start) {
    if (sum == W)
        print(include, size);
    else
        for (i = start; i < n; i++) {
            include[size] = Values[i];
            subset_sum(include, size + 1, sum + Values[i], i + 1);
        }
}
...
subset_sum(include, 0, 0, 0);

```

Giải thuật (cải tiến)

```

subset_sum(include[0 .. n-1], size, sum, start, total) {
    if (W == sum)
        print(include, size);
    else {
        int lost = 0;
        for (i = start; i < n; i++) {
            if ((sum + total >= W) && (sum + Values[i] <= W)) {
                include[size] = Values[i];
                subset_sum(include, size + 1, sum + Values[i], i + 1, total -
                Values[i] - lost);
            }
            lost += Values[i];
        }
    }
}
total =  $\sum_{i=0}^{n-1} Values[i]$ ;

```

```
if (Values[0] <= W && total >= W )  
    subset_sum(include, 0, 0, 0, total);
```