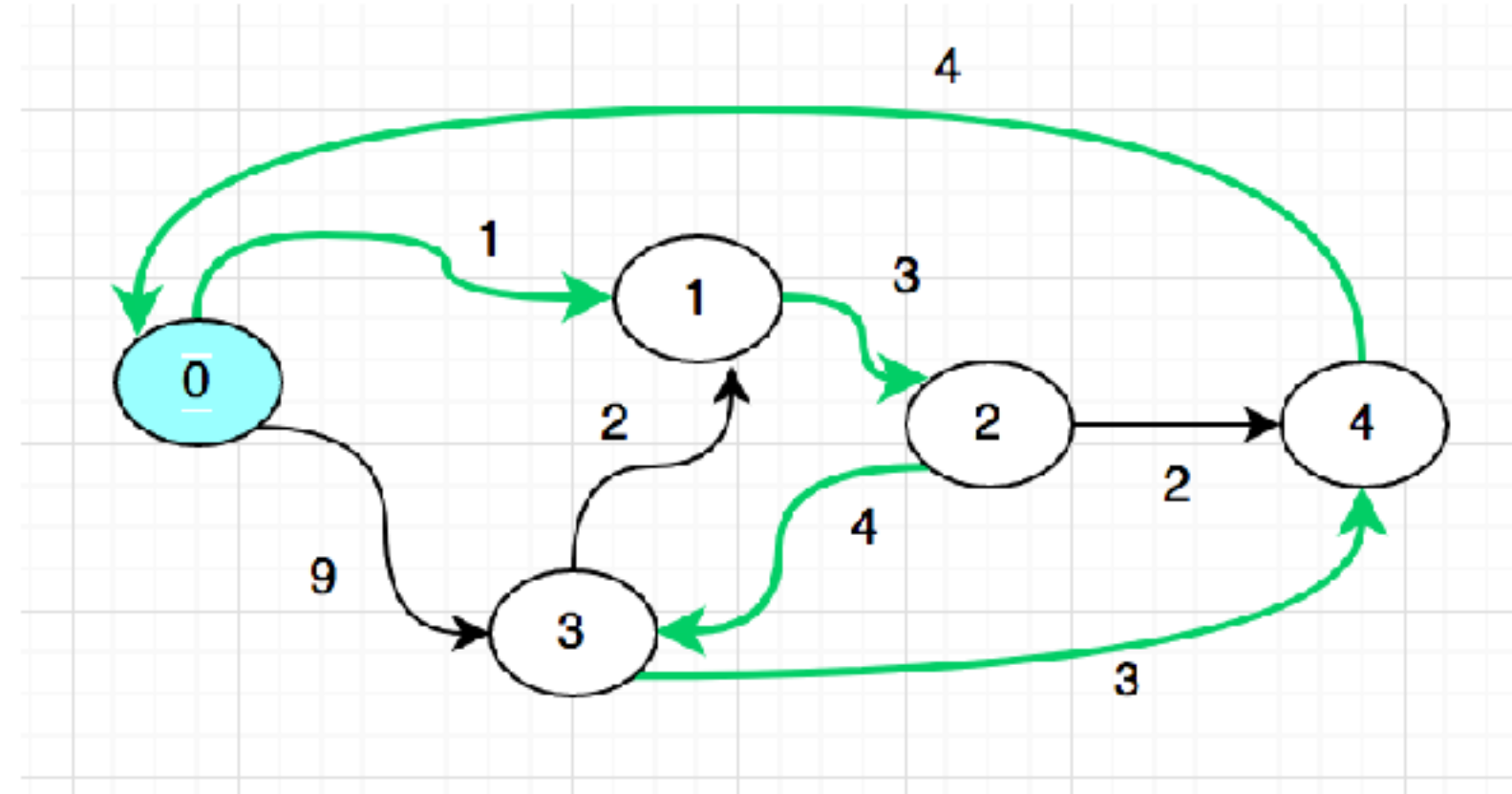


# Bitmask DP

What is Bitmasking ?

আজকে আমরা বিখ্যাত ট্রাভেলিং সেলসম্যান প্রবলেম বিটমাস্ক দিয়ে সলভ করবো। প্রবলেমটা হলো তোমাকে কিছু শহর এবং রাস্তা একটা গ্রাফ হিসাবে দেয়া আছে। তোমাকে প্রথম শহর থেকে শুরু করে সবগুলো শহর ঠিক একবার করে ভ্রমণ করে প্রথম শহরে ফিরে আসতে হবে। প্রশ্ন হলো সর্বনিম্ন কত দূরত্ব অতিক্রম করে তুমি কাজটা করতে পারবে?



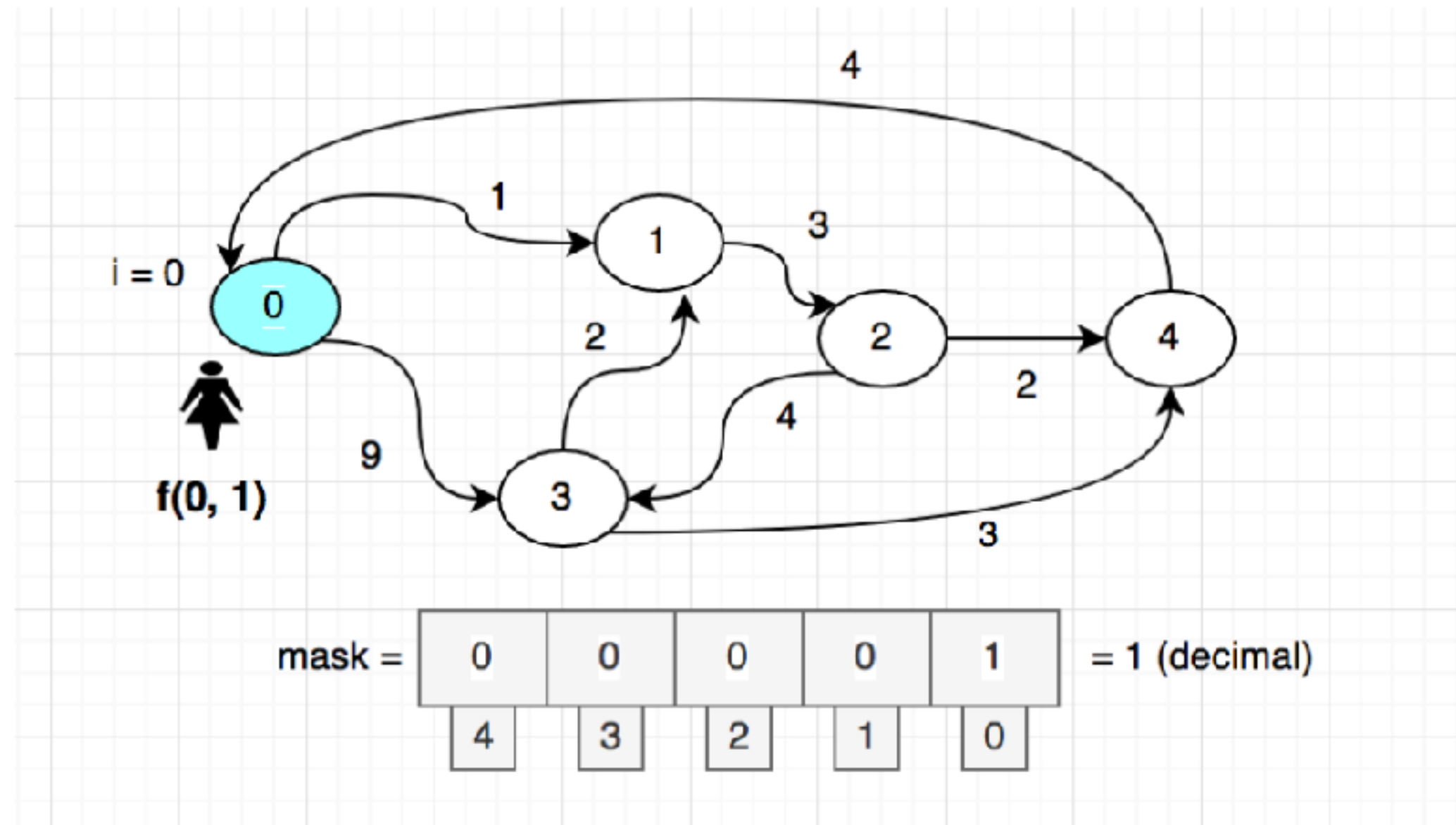
ছবির গ্রাফে  $n = 4$  টি শহর আছে এবং অপটিমাল পথ হলো  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 0$ , মোট দূরত্ব 15।

$DP[i]$  = Cost of completion if currently at  $i$ th city.

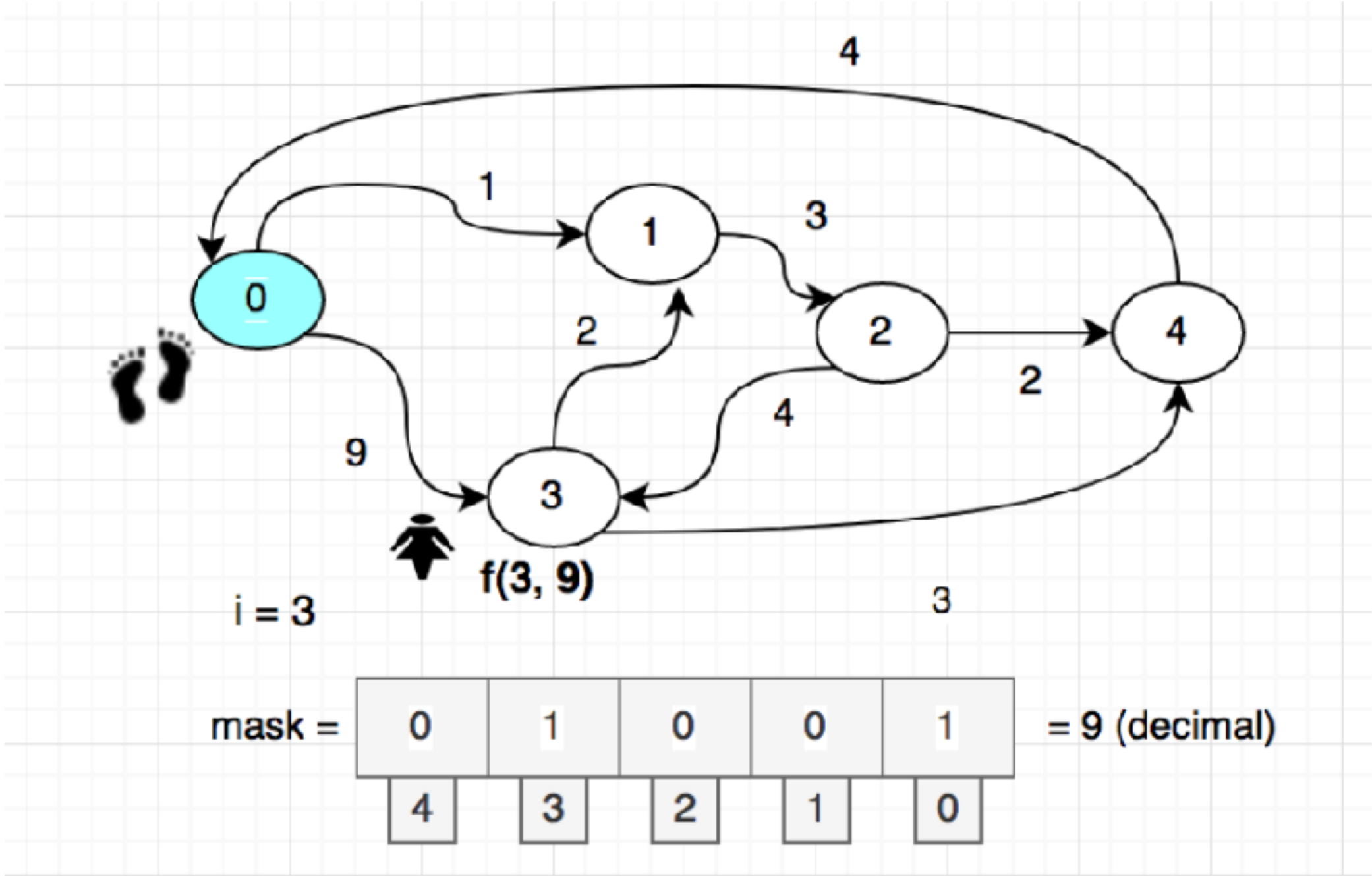
$DP[i]$  = Cost of completion if currently at  $i$ th city.

$DP[i][mask]$  = Cost of completion if currently at  $i$ th city and already visited some city. Visited cities are stored by mask.

উপরের উদাহরণে 5 টা শহর আছে, তাই বিট লাগবে 5 টা। আমরা ১টা ইন্টিজারের শেষ 5 টা বিট ব্যবহার করবো। শুরুতে আমরা শহর 0 তে আছি এবং শুধুমাত্র ওই শহরটাই ভ্রমণ করেছি। তাহলে মাস্ক হবে বাইনারি 00001, অর্থাৎ খালি 0 তম বিটটা অন থাকবে।



এখন যদি তুমি 0 থেকে 3 এ যাও তাহলে কি হবে?



এবার আমরা 3 নম্বর বিটটাকেও অন করে দিয়েছি। বাইনারি 01001 কে ডেসিমালে 9 লেখা যায়, তাই বর্তমান স্টেট হবে  $f(3, 9)$ ।

```
1 int turnOn(int x,int pos) {  
2     return x | (1<<pos);  
3 }
```

```
1 bool isOn(int N,int pos) {  
2     return (bool)(N & (1<<pos));  
3 }
```



রিকার্সিভ ফর্মুলাটা

$$f(i, 2^{n-1}) = dis[i][0]$$

$$f(i, mask) = \min(f(j, turnOn(mask, j) + w(i, j)) \text{ where } (i, j) \in E$$

এখানে  $w(i, j)$  দিয়ে শহর দুটির দূরত্ব বুঝিয়েছে।

```
int n;
int f(int i, int mask) {
    if (mask == (1<<n) - 1) {
        return w[i][0];
    }

    if (mem[i][mask] != -1) {
        return mem[i][mask];
    }

    int ans = INF;
    for (int j = 0; j < n; j++) {
        if (w[i][j] == INF) continue;

        if (isOn(mask, j) == 0) {
            int result = f(j, turnOn(mask, j)) + w[i][j];
            ans = min(ans, result);
        }
    }

    return mem[i][mask] = ans;
}
```



