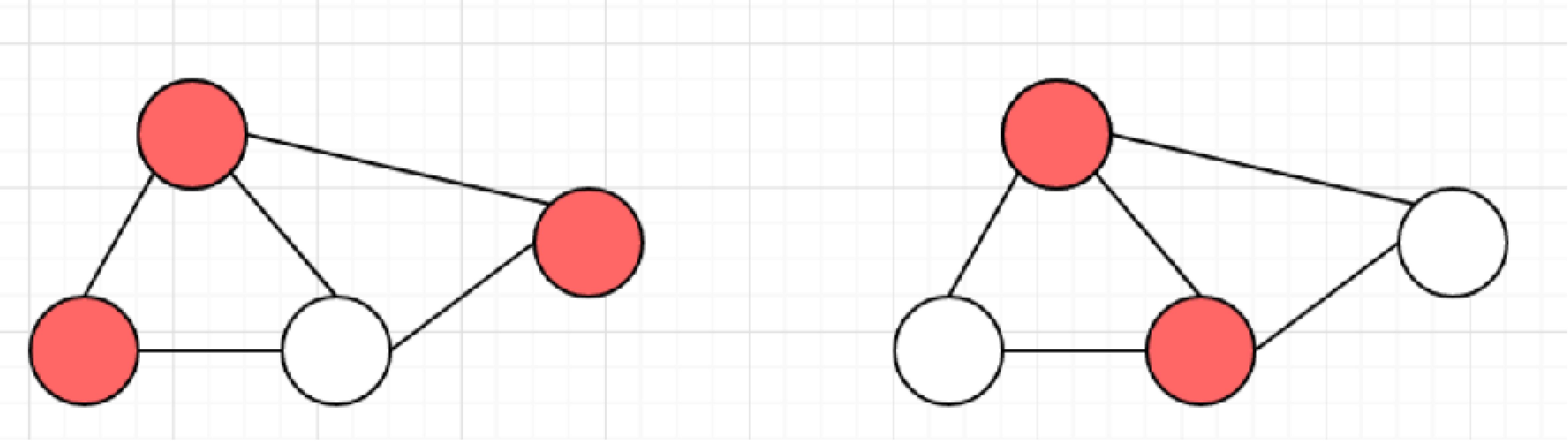
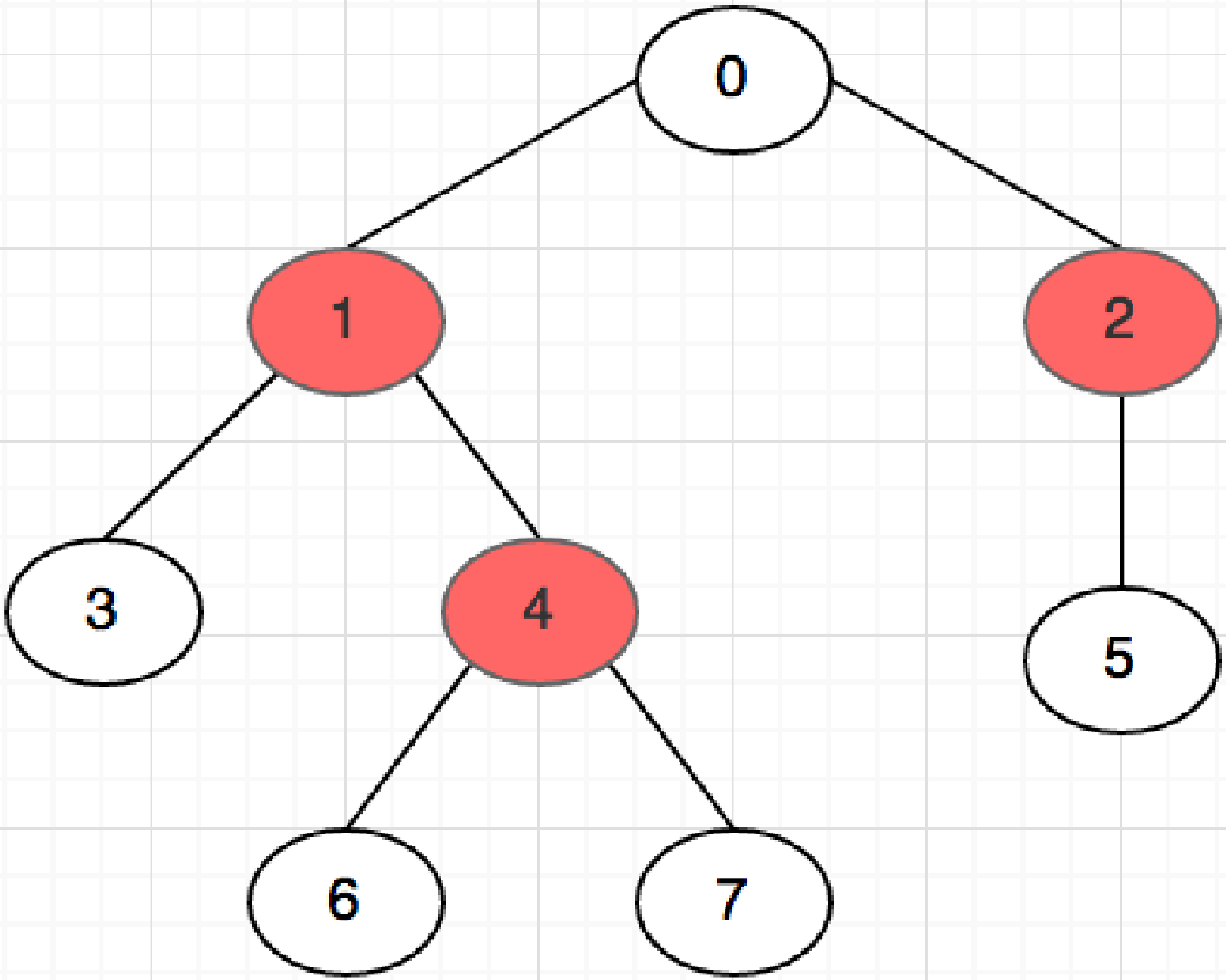
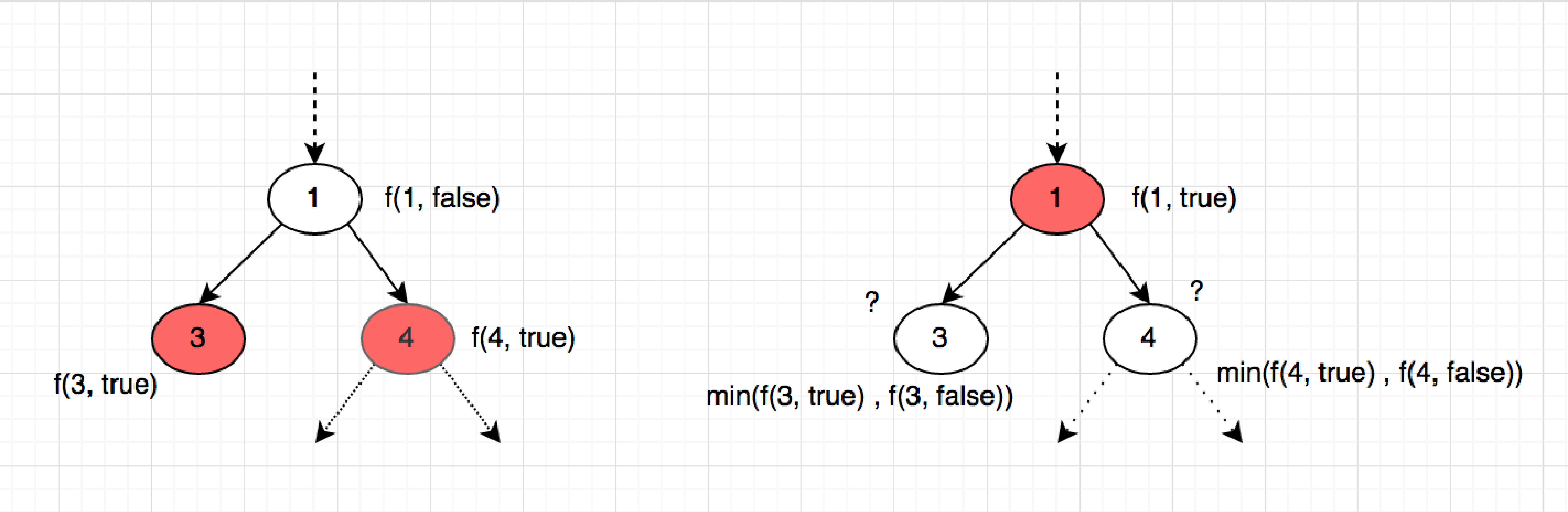
DP on Tree

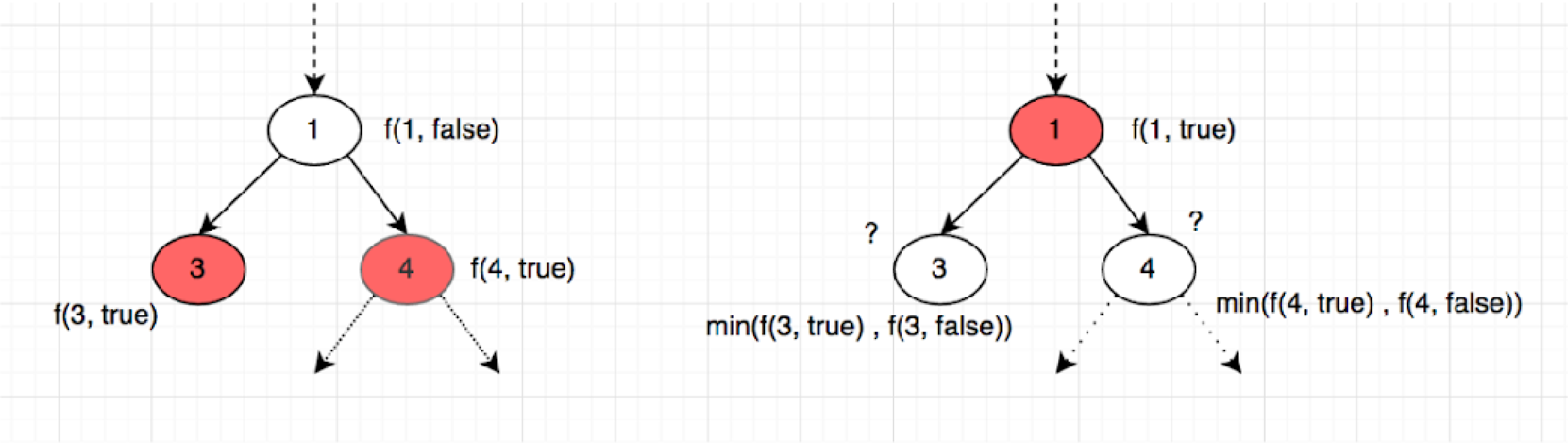
ধরা যাক একটি শহরে কিছু রাস্তা আছে, এখন প্রতি রাস্তায় মোড় বা জাংশনে আমরা পাহারাদার বসাতে চাই। কোনো মোড়ে পাহারাদার বসালে সে মোড়ের সাথে যুক্ত রাস্তাগুলো একাই পাহারা দিতে পারে। এখন তোমাকে বলতে হবে সব কয়টা রাস্তা পাহারা দিতে নূন্যতম কয়জন পাহারাদার দরকার?

একটা উদাহরণ দেখি। নিচের শহরে লাল নোডগুলোতে পাহারাদার বসানো হয়েছে:




0
1
2
3
4
5
6
7


1
f(1, false)
3
f(3, true)
4
f(4, true)
1
f(1, true)
?
3
min(f(3, true) , f(3, false))
?
4
min(f(4, true) , f(4, false))



এখন যদি 1 এ পাহারাদার না বসাই তাহলে আমাদেরকে $f(1, false)$ সমস্যাটার সমাধান করতে হবে। উপরে বামের ছবিতে সেটাই দেখানো হয়েছে। যেহেতু 1 এ পাহারাদার নাই, নিচের রাস্তা দুটি গার্ড দেয়ার জন্য 1 এবং 2 তে পাহারা বসাতেই হবে এবং $f(3, true)$ আর $f(4, true)$ সাবপ্রবলেমদুটো সলভ করতে হবে:

$$f(1, false) = f(3, true) + f(4, true)$$

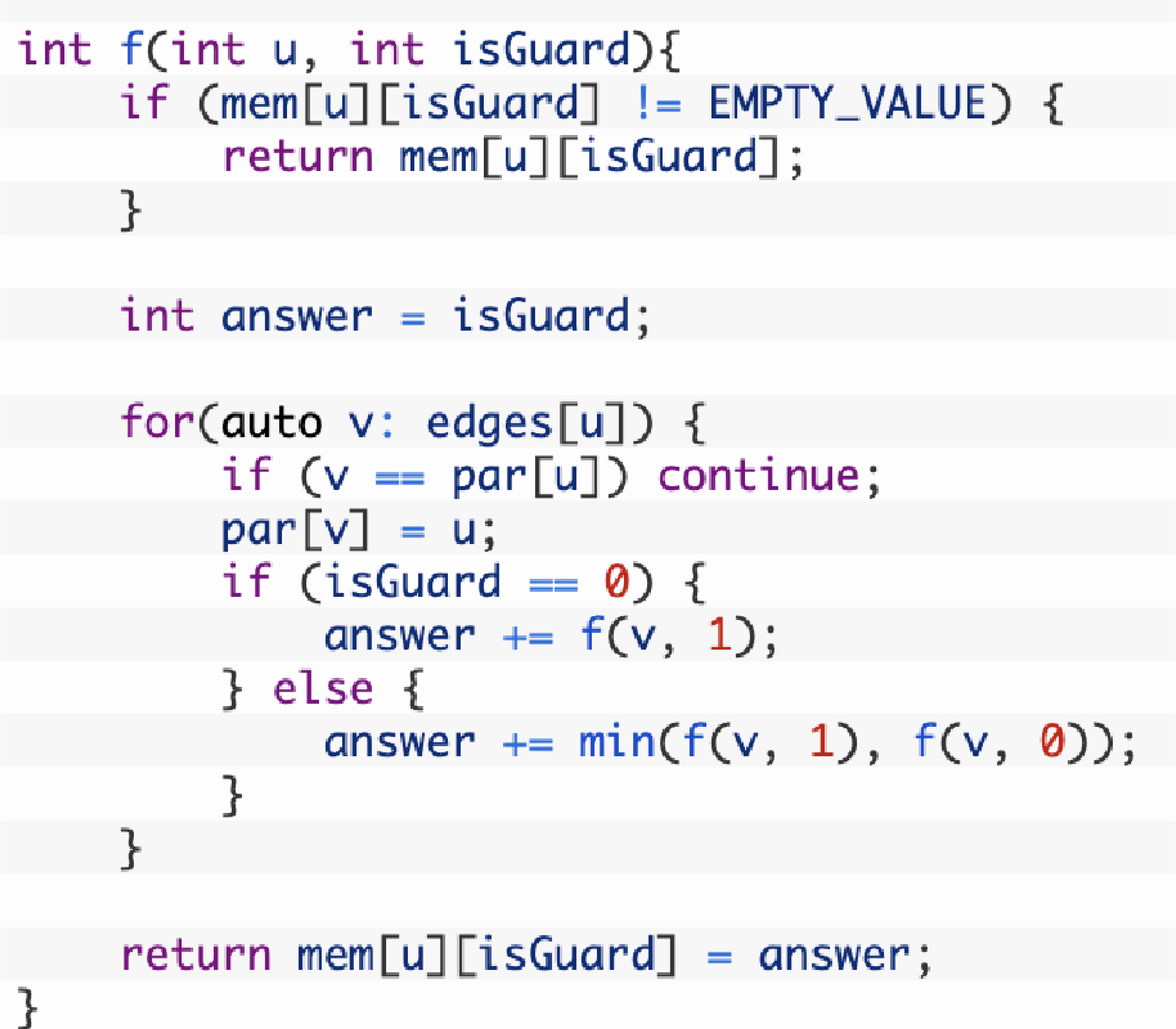
আবার যদি 1 এ পাহারা বসাই তাহলে 3 এবং 4 এ পাহারাদার বসানো অপশনাল। আমরা একবার বসিয়ে এবং একবার না বসিয়ে সবপ্রবলেমগুলো সলভ করে দেখবো কোনটা অপটিমাল হয়:

$$f(1, true) = 1 + min(f(3, true), f(3, false)) + min(f(4, true) + f(4, false))$$

$$f(u,0) = \sum f(v,1)$$

$$f(u,1) = 1 + \sum min(f(v,1),\ f(v,0))$$

$$where\ (u,v) \in E$$



```
int f(int u, int isGuard){
    if (mem[u][isGuard] != EMPTY_VALUE) {
        return mem[u][isGuard];
    }

    int answer = isGuard;

    for(auto v: edges[u]) {
        if (v == par[u]) continue;
        par[v] = u;
        if (isGuard == 0) {
            answer += f(v, 1);
        } else {
            answer += min(f(v, 1), f(v, 0));
        }
    }

    return mem[u][isGuard] = answer;
}
```