

ICPC2021 模擬国内予選 G 問題 クッキーゲーム

原案: tsutaj

問題文: climpet

データセット: tsutaj

解答: tsutaj・climpet

解説: tsutaj

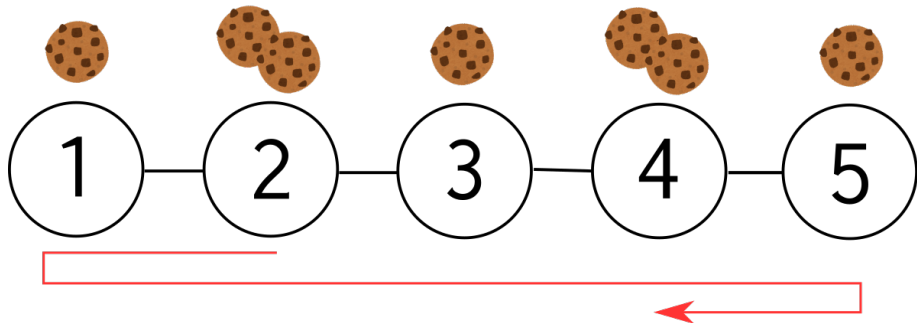
2021 年 10 月 24 日

クッキーゲーム

- ▶ N 頂点からなる無向木が与えられる。各頂点には 1 枚または 2 枚のクッキーがある
 - ▶ 木の頂点数 N : $2 \leq N \leq 10^5$
- ▶ 木の頂点を 1 つ選び、そこに駒を置いたのち、以下を繰り返す
 - ▶ 現在駒が置かれている頂点にあるクッキーを 1 枚だけ食べる
 - ▶ 駒が置かれている頂点に隣接する頂点を 1 つ選び、そこに駒を動かす
 - ▶ 移動先の頂点にクッキーが 1 枚もない場合、その時点で操作は終了
- ▶ 最適に行動したとき、最大で何枚のクッキーを食べられるか？

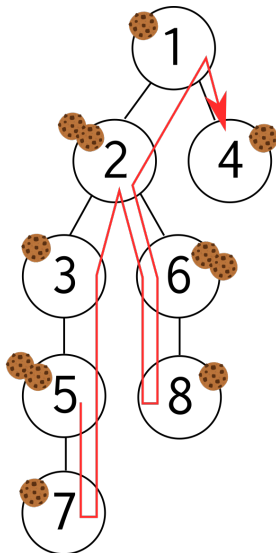
例

入力例 2 に対応するもの



例

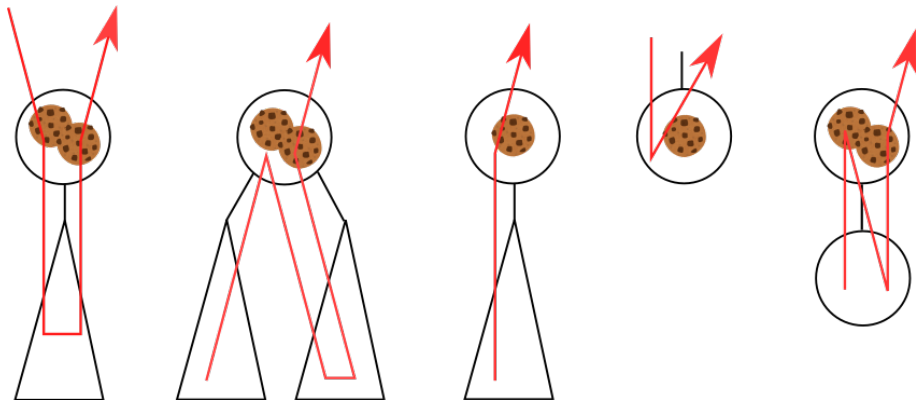
入力例 4 に対応するもの



- ▶ ある 1 頂点を根とみなし、根付き木として考える
- ▶ 頂点 v を根とする部分木について、 v をどのように使用したかで場合分けし動的計画法 (DP) を適用する
- ▶ 状態の持ち方の一例
 - ▶ 使用回数が余っており、親方向に繋がられる場合
 - ▶ v で折り返し、親方向の要素は使わない場合
- ▶ 適切に実装すれば $O(N)$ で実行可能

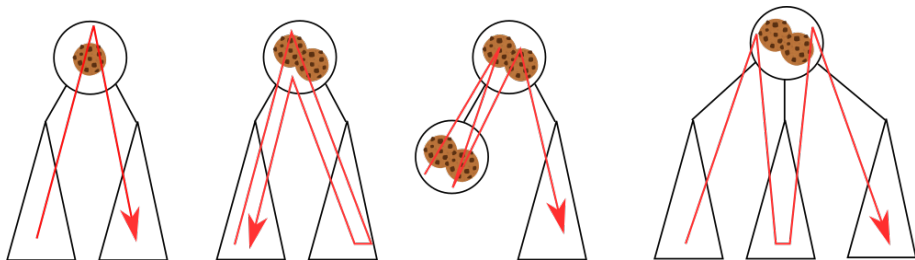
考慮すべきケースの例 (1)

- ▶ 使用回数が余っており、親方向に繋げられる場合の例



考慮すべきケースの例 (2)

- ▶ 部分木の根 v で折り返し、親方向の要素は使わない場合の例



ICPC で戦う上での tips: 全探索を書こう

- ▶ コーナーケースが多めの問題だが全探索はすぐ書けるので、全探索解と比較してコーナーとなる入力を見つけよう
 - ▶ $O(N^3)$ なら比較的簡単に用意でき、 $N \leq 20$ なら手元で十分動く

```
int ans = 0;
vector<int> vec, ans_vec;
auto dfs = [&](auto &&self, int cur, int step) -> void {
    // 答えが更新された
    if(ans < step) {
        ans = step;
        ans_vec = vec;
    }
    // 隣接する頂点に移動
    for(auto to : G[cur]) {
        // クッキーがなければ移動不可
        if(nums[to] == 0) continue;
        nums[to]--; vec.emplace_back(to + 1);
        self(self, to, step + 1);
        nums[to]++; vec.pop_back();
    }
};
for(int i=0; i<N; i++) {
    nums[i]--; vec.emplace_back(i + 1);
    dfs(dfs, i, 1);
    nums[i]++; vec.pop_back();
}

printf("%d\n", ans);
// 答えについて、その経路を出すと便利
#ifdef DEBUG
for(size_t i=0; i<ans_vec.size(); i++) {
    cout << ans_vec[i];
    if(i + 1 < ans_vec.size()) cout << " -> ";
    else cout << endl;
}
#endif
```


▶ Writer 解

- ▶ tsutaj (C++・208 行・8648 bytes)
- ▶ climpet (C++・173 行・3245 bytes)

▶ 統計

- ▶ AC / tried: 8 / 14 (57.1 %)
- ▶ First AC
 - ▶ The University of Tokyo: UT a.k.a Is (51 min 30 sec)