

COOKIE READING MACHINE

問題: drafear

解答: drafear, asi1024, n_knuu, amano

解説: drafear

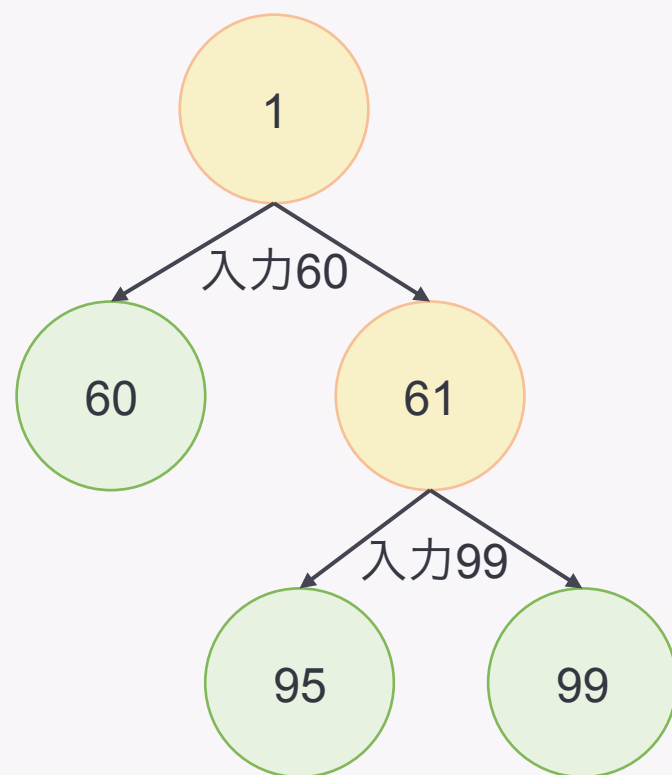
問題概要

- 最初, 美味しさ D ($1 \leq D \leq 127$) のクッキーが1枚ある
- クッキー☆増殖装置にクッキーを1枚入れて(美味しさを d とする) 数字 x ($0 \leq x \leq 127$) を入力すると
美味しさ x のクッキーと, 美味しさ $(x \text{ XOR } d)$ のクッキーになる
- クッキー☆増殖装置をちょうど $N-1$ 回使用して生成された N 枚のクッキーの美味しさの和を最大化したい



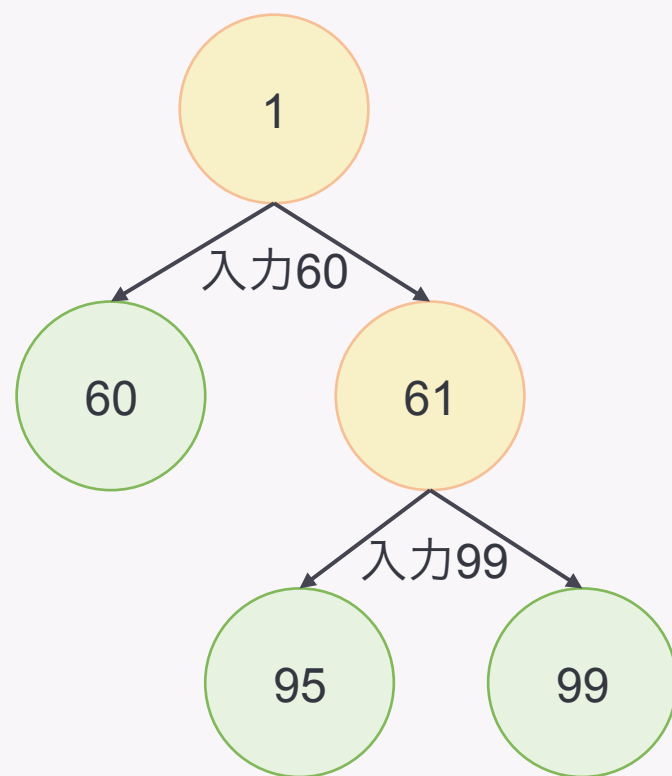
例

- 美味しさ ($D=$)1のクッキーから ($N=$)3枚のクッキーを作る



例

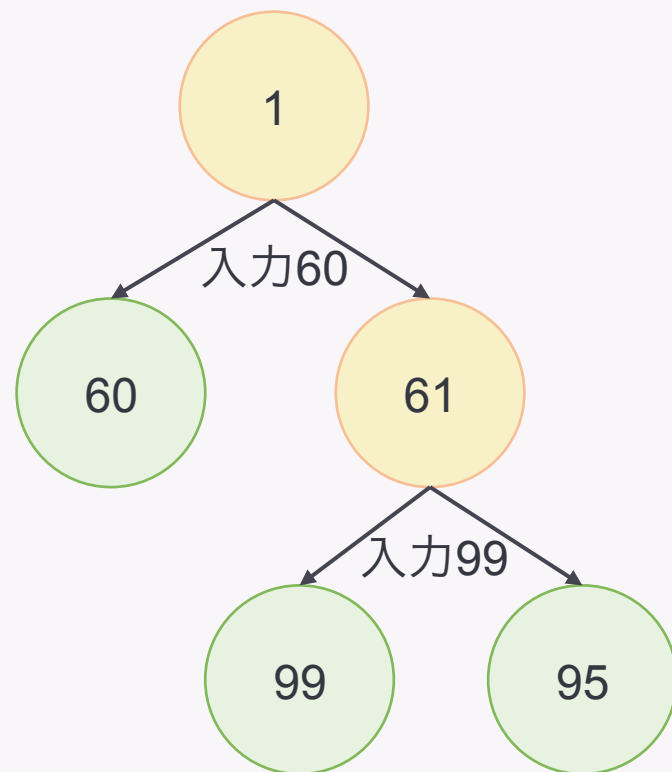
- この場合が最大で, $60+95+99=255$ を出力する



考察

考察

- 二分木になっていて, 左の子 XOR 右の子 $= x XOR (x XOR d) = d$ (親)
- これを根まで辿ると, 根(D)=葉1 XOR 葉2 $XOR \dots XOR$ 葉N



考察

- つまり, 次の問題と等価である
 - $a_1 + a_2 + \dots + a_N$ を最大化したい
 - ただし, $a_1 \text{ XOR } a_2 \text{ XOR } \dots \text{ XOR } a_N = D$ であって, $1 \leq a_i \leq 127$ ($1 \leq i \leq N$)
- a_1, \dots, a_N を決めれば, これらを葉とし, 適当に繋げれば装置の使用手順が復元できる
- XORはビットごとなので, ビットごとに独立して考えて良い

解法

解法

- N が奇数の場合
 - 最後のビットが1なら1を N 回立てる
 - 最後のビットが0なら1を $N-1$ 回立てる
- N が偶数の場合
 - 最後のビットが1なら1を $N-1$ 回立てる
 - 最後のビットが0なら1を N 回立てる

解法

- 例の場合

| | おいしさ(2進数) | おいしさ(10進数) |
|-----------|-----------|------------|
| 元となるクッキー | 0000001 | 1 |
| 最終生成物 1枚目 | 1111111 | 127 |
| 最終生成物 2枚目 | 1111111 | 127 |
| 最終生成物 3枚目 | 0000001 | 1 |

- 計算量は, データセット数 T を考慮すると, $O(T \log D)$

解法

- 127が実際大半を占める
- 実は, 次のようにしても良い
 - N が奇数の場合, $127(N-1)+D$
 - N が偶数の場合, $127(N-1)+(D \text{ XOR } 127)=127N-D$
- 時間計算量は $O(T)$

解法

- $dp[i][j] := (a \downarrow 1 \text{ XOR } \dots \text{ XOR } a \downarrow i = j \text{ となる ときの } a \downarrow 1 + \dots + a \downarrow i \text{ の 最大値})$ としたDPでも解ける
- 前処理しておけば, 各クエリに対して $O(1)$ で答えられるので, 計算量は $O(T + ND^2)$

統計・ジャッジ解

統計

- First Accept
 - ALL 🍷(^ ^)🍷←寿司をにぎっています (05:36)
 - Onsite(東京) 🍷(^ ^)🍷←寿司をにぎっています (05:36)
 - Onsite(京都) ixmel (24:46)
- Accept 152 / 184 (83%)

ジャッジ解

- drafeear
 - $O(T)$, C++, 23行 431B
 - $O(T \log D)$, C++, 31行 579B
 - dp, C++, 50行 1034B
 - dp, js, 74行 1958B
- n_knuu
 - $O(T)$, Python, 15行 309B
- amano
 - $O(T)$, C++, 29行 429B
- asi1024
 - dp, C++, 33行 609B