みんなのプロコン 2018 予選 解説

2018 年 2 月 10 日

A: yahoo

文字列を読み込み、その 0, 1, 2 文字目 (ただし、添え字は 0 から開始) が順に y,a,h であるかと、3, 4 文字目が等しいかどうかを判定すればよいです。

```
#include<stdio.h>
int main()
{
        char s[10];
        scanf("%s", s);
        if (s[0] == 'y'&&s[1] == 'a'&&s[2] == 'h'&&s[3] == s[4])printf("YES\n");
        else printf("NO\n");
}
```

B: オークション

Y の下 K 桁が 0 であることは、Y が 10^K の倍数であることと同値です。$X+1$ 以上の最小の K の倍数を出力すればよく、それは以下のように求められます。

```
#include<stdio.h>
int main()
{
        int x, k;
        scanf("%d%d", &x, &k);
        int t = 1;
        for (int i = 0; i < k; i++)t *= 10;
        printf("%d\n", (x + t) / t*t);
}
```

C：駆引取引

こういった 2 人対戦ゲームに関する問題ではそれぞれのプレイヤーの戦略を実際に考えてみることが有効です。高橋君と青木君の戦略は以下のように書けます。

- 高橋君の戦略：売却を行った方が最終的な得点が高くなるならば、売却を行う。そうでなければ商品を購入する。
- 青木君の戦略：販売を停止したときに、最終的な得点が最も低くなるような商品を選んで販売を停止する。

どのような順序で商品が廃止されたかは重要ではなく、今現在どの商品の販売が停止しているか、だけが重要であることは明らかでしょう。ここから、今販売が停止されている商品を状態とした動的計画法により最終的な得点を求めることを考えてみます。

S を商品の集合、$dp(S)$ を S に含まれる商品の販売が停止しているときの最終的な得点、$f(S,k)$ を S に含まれる商品の販売が停止しているときに、k 個財宝を売却したときの所持金で購入を行ったときの得点の最大値、とします。先程の観察に基づくと、$dp(S)$ は以下のようにして求められます。

$$dp(S) = \max(f(S,|S|), \min_{t \notin S}\{dp(S \cup \{t\}))$$

さて、$f(S,k)$ (特に $f(S,|S|)$ の具体的な値さえ分かれば、この問題を解くことが可能なことが分かりました。当たり前ですが、高橋君は今残っている商品を全て購入可能ならばそうするべきです。そうでない場合を考えます。このとき、$f(S,k) = f(S \cup \{t\}, k), t \notin S$ を満たす t が必ず存在するはずです。よって、$t \notin S$ を満たす t について全探索を行い、$f(S \cup \{t\}, k)$ の最大値を求めればよいです。こちらも動的計画法を用いて求めることができます。

全体として、$O(2^N N^2)$ の計算量で答えを求めることが可能なことが分かりました。$N \leq 18$ より、これは十分高速に動作します。

D : XOR XorY

この解説においては、$\oplus$ をビット XOR の記号とします。また、数列に含まれうる数の最大値を M とします。

まずはじめに、良い数列としてありうる数列は何通りか？という問題について考えます。

簡単のため、$X = Y = 0$ の場合を考えてみます。このとき，$A_{i,j} = a_i \oplus a_j$ という形の制約を満たすような a はあるか？という問題になります。XOR 演算の性質より、$a_i \oplus a_j = a_k \oplus a_i \oplus a_k \oplus a_j$ が成立します。よって、$A_{i,j} = A_{1,i} \oplus A_{1,j}$ が全ての (i,j) について成立するかどうかを確認することで良い数列が存在するかどうかを調べることができます。さらに、a_1 の値を固定してみると、$a_2, a_3, \dots, a_N$ は A から一意に定まります。

$X \neq Y$ のみを満たす場合を考えてみましょう。先程と同様に考えていくことにします。$A_{1,i} \oplus A_{1,j}$ が $a_i \oplus a_j$ あるいは $a_i \oplus a_j \oplus X \oplus Y$ のどちらかであることから、$A_{i,j} = A_{1,i} \oplus A_{1,j} \oplus X$ あるいは $A_{i,j} = A_{1,i} \oplus A_{1,j} \oplus Y$ のどちらかを満たす必要があることが分かります。さらに、a_1 の値を固定してみると、$a_2, a_3, \dots, a_N$ の値はほぼ一意に定まります。より正確には $i > 1$ を満たす i について、a_i は $A_{1,i} \oplus a_1 \oplus X$ か $A_{1,i} \oplus a_1 \oplus Y$ のどちらかであることが分かります。

ここまでで、良い数列が存在するかどうかを判定し、良い数列としてありうる数列の数を数えることが $O(K(K+M))$ で出来ました。以降では、元の問題について考えていきます。

ここまでの観察から、a が良い数列であったとき a_i を $a_i \oplus X \oplus Y$ に置き換えた数列も良い数列である、という性質に気づくかもしれません。これは、$A_{i,j} = a_i \oplus a_j \oplus X$ としたとき、$A_{i,j} = a_i \oplus X \oplus Y \oplus a_j \oplus X = a_i \oplus a_j \oplus Y$ となることから明らかです。この性質からある整数 x と $x \oplus X \oplus Y$ はペアとなっていることが分かります。すると、a_1 がどのペアに属するかを決めたとき、$a_2, \dots, a_N$ がどのペアに属するかが定まる、と考えることができます。各ペアごとに着目してみると、以下のような問題となっていることが分かります。

> 文字 `A` が a 個、文字 `B` が b 個ある。これら $a+b$ 個から n 個選んで並べて作ることのできる文字列は何種類あるか？

$a \le b$ を仮定しても一般性を失いません。この問題の答えは、$\sum_{i=\max(0,n-b)}^{\min(a,n)} \binom{n}{i}$ となります。二項係数を前もって $O(M^2)$ で計算しておくことで、a_1 がどのペアに属するかを固定したときの答えをならし $O(M+K)$ で求めることができます。

以上より、この問題は $O(N+M(M+K)+K^2)$ で解くことが可能です。$N, M \le 2048, K \le 1024$ より十分高速に動作します。

E: グラフの問題

Erdos-Gallai theorem と呼ばれる以下の定理が、この問題を解くのに有用です。

定理

次数 $d_1 > d_2 > \cdots d_N$ の単純グラフが存在する条件は、以下で与えられる。

- d_i たちの合計は偶数
- 全ての k に対し、$sum_{i=1}^{k} d_i - k(k-1) \leq \sum_{i=k+1}^{N} min(d_i, k)$ が成り立つ

この問題を解くにあたって証明の必要はありませんが、この条件の必要性にのみ言及しておきましょう。1 つめの条件は、各辺が 2 回ずつ数えられることに依ります。2 つめの条件は、最初の k 点とその他の点を分けるカットを考えたとき、最初の k 点からその外へ最低でも $sum_{i=1}^{k} d_i - k(k-1)$ 本の辺が出るが、その他の点は最初の k 点から伸びる辺を高々 $\sum_{i=k+1}^{N} min(d_i, k)$ 本しか受け入れられないことに依ります。

十分性については、この証明 (https://www.sciencedirect.com/science/article/pii/S0012365X09004683?via%3I が分かりやすいでしょう。

さて、この定理を用いれば、適切な実装により「答えが Yes かどうか」は判定できます。また、1 つめの条件が満たされ、かつ答えが Yes でない場合、答えは Absolutely No です。

答えが No かどうかはどのように判定すればいいでしょうか。実は、d_i たちの中で最大の要素をひとつ選んで 1 減らした列について上記の定理を適用すれば、その判定も可能です。これは、ある d_i を 1 増やしてグラフが存在するなら、かわりに d_i を増やした頂点と辺で結ばれる頂点ひとつの d_i の値を 1 減らしてもグラフが存在すること (およびその逆) と、上記の定理の 2 番目の条件を greedy に考察することによってわかります。