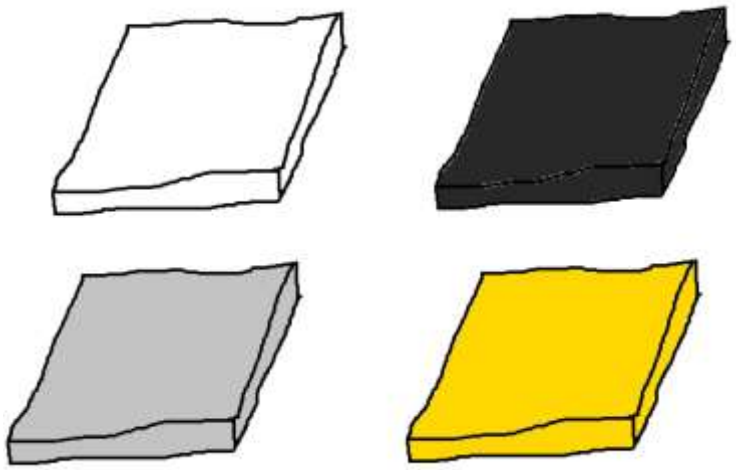
DDCC2018 予選解説

Writer: E869120, square1001

2018 年 11 月 23 日 21:00 ～ 22:30



問題 A: チップ・ストーリー ～無色編～

それぞれの操作で、チップがそれぞれ 4 つの小さいチップに分かれます。したがって、DIVCO 君が 1 回の処理を行う前のチップの個数が X 枚のとき、処理を行った後のチップの枚数は $X \times 4$ 枚となります。

この問題には 3 つの解法があります。それぞれについて説明していきます。

☆ 解法 1 – ループを使って繰り返し

現在のチップの枚数を X とします。最初、$X = 1$ で、1 回処理を行うときに X が 4 倍されます。つまり、「X を 4 倍する (プログラミングでは「$X = X \times 4$」という操作)」を N 回繰り返して、最終的なチップの枚数が X になります。

「N 回繰り返す」は、C++, Java, Python をはじめとする多くのプログラミング言語では for 文を使って実装することができます。知らなかった人は調べてみましょう。

サンプルコード (C 言語): https://beta.atcoder.jp/contests/ddcc2019-qual/submissions/3635339

☆ 解法 2 – if 文で $N = 1, 2, 3, 4, 5$ のときに場合分け

$N \leq 5$ なので、手計算でも答えを出すことができます。答えは以下のようになります。

- $N = 1$ のとき、4 枚
- $N = 2$ のとき、16 枚
- $N = 3$ のとき、64 枚
- $N = 4$ のとき、256 枚
- $N = 5$ のとき、1024 枚

多くのプログラミング言語では if-else 文を使って条件分岐できます。入力される N の値によって条件分岐し、出力する値 (あるいは “文字列”) を変えることによってこの問題を解くことができます。

サンプルコード (C 言語): https://beta.atcoder.jp/contests/ddcc2019-qual/submissions/3635348

☆ 解法 3 – シフト演算子を使う

少し考えれば、この問題の答えは 4^N (4 を N 回掛けた値) になることが分かります。実は、これは 2^{2N} (2 を $2N$ 回掛けた値) と同じです。

多くのプログラミング言語では、「シフト演算子」という演算子があります。C++ や Java でいう “<<” がまさにそれです。”a << x” は、$a \times 2^x$ と同じなので、この問題の答えは “1 << $2x$” で求めることができます。

サンプルコード (C 言語): https://beta.atcoder.jp/contests/ddcc2019-qual/submissions/3635353

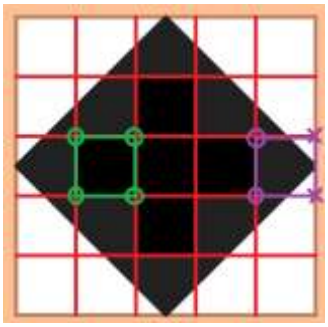
B – チップ・ストーリー　～漆黒編～

この問題には、二通りの解法があります。今回は、それぞれの解法について、図を用いて分かりやすく説明したいと思います。

☆ **解法 1: それぞれのマスに対して判定**

マスは合計で N^2 個ですので、制約の $N \leq 100$ であれば、セルごとに判定して数えることは十分に可能です。さて、どのような条件を満たせば黒いマスになるのでしょうか？

結論から言うと、「四隅が全て黒ければ (= チップに含まれれば)、そのマスは黒いマス」です。何故なら、チップは直線で構成されている、曲線を含まない正方形だからです。



例えば、上図だと緑のマスは四隅全てがチップに含まれるため黒いマスですが、紫のマスは右側二個の済がチップに含まれないため、白いマスです。

そこで、画像全体を (2 倍拡大して) 大きさ $2N \times 2N$ の正方形として考えます。黒い正方形のチップの中心座標は (N, N) となります。また、ある座標 (x, y) がチップに含まれている条件は $|x - N| + |y - N| \leq N$ です。「マンハッタン距離」(※注参照) が N 以下ということです。

ですので、上から i 番目 $(1 \leq i \leq N)$、左から j 番目 $(1 \leq j \leq N)$ のマスが黒いマスである条件は、以下の全てを満たすことです。

1. 隅 $(2i - 2, 2j - 2)$ に対して： $|N - (2i - 2)| + |N - (2j - 2) \leq N$
2. 隅 $(2i - 2, 2j)$ に対して： $|N - (2i - 2)| + |N - (2j) \leq N$
3. 隅 $(2i, 2j - 2)$ に対して： $|N - (2i)| + |N - (2j - 2) \leq N$
4. 隅 $(2i, 2j)$ に対して： $|N - (2i)| + |N - (2j) \leq N$

この条件で、N^2 個全てのマスについて数え上げればこの問題を解くことができます。

サンプルコード (C++)： https://beta.atcoder.jp/contests/ddcc2019-qual/submissions/3635345

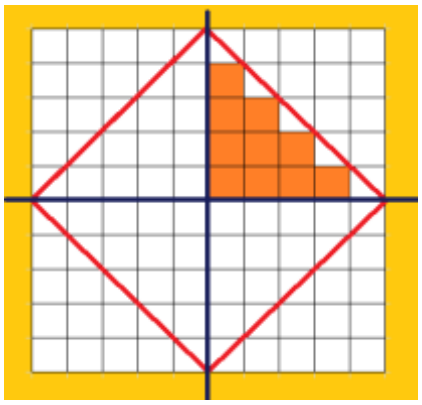
注： 座標 (a, b) と (c, d) のマンハッタン距離は、$|a - c| + |b - d|$ で表される。

(この問題の解説は、次ページに続いています)

☆ **解法 2: 計算式によって一発で答えを求める**

この解法では、N が偶数の場合と奇数の場合とで場合分けします。

A. N が偶数の場合

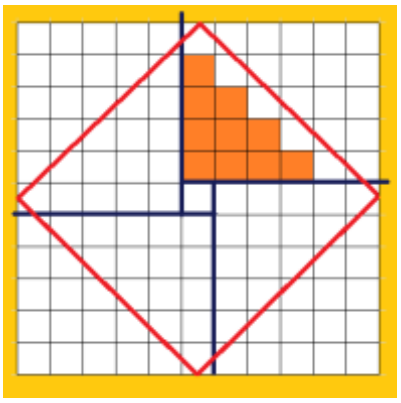


画像を右図の青線のように真ん中で四分割することを考えます。四分割された画像は合同ですので、全体の答え A はそのうちの一つの答え B の四倍となります。

さて、右図のオレンジの部分の個数が B となりますが、これは $1+2+3+4=10$ 個、という形になっています。N に対して一般化すると、$1+2+3+\cdots+\left(\frac{N}{2}-1\right)$ となります。

したがって、$A=4\times\left(1+2+3+\cdots+\left(\frac{N}{2}-1\right)\right)=2\times\frac{N}{2}\times\left(\frac{N}{2}-1\right)$ となります。

B. N が奇数の場合



画像を右図の青線のように五分割することを考えます。五分割のうち真ん中の一マスを除く四つに関しては、パターン A と同じように合同となっています。

また、右図のオレンジの部分の個数（分割されたうちの一つに対する黒いマスの個数の合計）は、これもまたパターン A と同じように $1+2+3+4=10$ 個という形になっています。ですので、全体の黒いマスの個数は $4\times10+1=41$ 個というように計算できます。N に対して一般化すると、$2\times\frac{N-1}{2}\times\left(\frac{N-1}{2}-1\right)+1$ となります。

そのように、数学的な手法を用いて $O(1)$ で計算することも可能です。

サンプルコード（C++）： https://beta.atcoder.jp/contests/ddcc2019-qual/submissions/3633732

問題 C: チップ・ストーリー ～白銀編～

すべての (i, j) に対して $P_i \times Q_j$ が 1 以上 N 以下となっている条件は次のようになります。

> $P_{max} = \max(P_1, P_2, \ldots, P_{10})$, $Q_{max} = \max(Q_1, Q_2, \ldots, Q_{10})$ として、$P_{max} \times Q_{max} \leq N$

なぜこれが条件となるのでしょうか。まず、$P_1, P_2, \ldots, P_{10}. Q_1, Q_2, \ldots, Q_{10}$ はすべて 1 以上の整数にするのが前提なので、どの 2 つを掛け算しても 1 以上になります。また、$P_i \times Q_j$ が最も大きくなるのは、最も大きい P_i を選び、最も大きい Q_j を選んだときなので、このときの $P_i \times Q_j$ は $P_{max} \times Q_{max}$ となるからです。

つまり、$P_{max} \times Q_{max} \leq N$ となる $(P_1, P_2, \ldots, P_{10}, Q_1, Q_2, \ldots, Q_{10})$ の組み合わせの個数を $(1\,000\,000\,007$ で割った余りを) 求めればよいことになります。

$P_{max} \leq c$ と決まっているとき、$(P_1, P_2, \ldots, P_{10})$ の組み合わせは何通りあるでしょうか？答えは c^{10} 通りです。なぜなら、$P_1, P_2, \ldots, P_{10}$ それぞれが 1 以上 c 以下となればよいから、それぞれについて c 通りずつ選択肢があるからです。

$P_{max} = c$ と決まっているときの $(P_1, P_2, \ldots, P_{10})$ の組み合わせの個数は、$P_{max} \leq c$ のときの組み合わせの個数に、$P_{max} \leq c - 1$ の組み合わせの個数を引いた値と同じになるので、$c^{10} - (c-1)^{10}$ 通りであることが分かります。

例えば、$P_{max} = 5$ となる $(P_1, P_2, \ldots, P_{10})$ の組み合わせの個数は何通りでしょうか？$P_{max} \leq 5$ となるものは 5^{10} 通りあります。なぜなら、すべての P_i が 1, 2, 3, 4, 5 のどれかだから、5 通りの選択肢があるからです。同様にして、$P_{max} \leq 4$ となるものは 4^{10} 通りあります。これより、$P_{max} = 5$ となるものは $5^{10} - 4^{10} = 8\,717\,049$ 通り、と求められます。

そこで、P_{max} を決め打ちしてから、$P_{max} \times Q_{max} \leq N$ となる場合の数を求めることを考えます。$P_{max} = c$ のとき、$Q_{max} \leq floor(N/c)$ とならなければならない (ただし $floor(x)$ は x を整数に切り捨てた値) です。なので、$(P_1, P_2, \ldots, P_{10}, Q_1, Q_2, \ldots, Q_{10})$ の組み合わせの個数は・・・

$(c^{10} - (c-1)^{10}) \times floor(N/c)^{10}$ です！あとは、全探索するだけです。P_{max} は 1 以上 N 以下の範囲にしぼられるので、$P_{max} = c$ を全探索して、それぞれの c に対して先ほどの式どおり組み合わせの個数を計算して、この合計がこの問題の答えになります。

$(c^{10} - (c-1)^{10}) \times floor(N/c)^{10}$ は計算量 $O(1)$ で計算できるので、全体の計算量 $O(N)$ でこの問題を解くことができました！

サンプルコード (C 言語) : https://beta.atcoder.jp/contests/ddcc2019-qual/submissions/3635865

(この問題の解説は次ページに続きます)

☆ **おまけ －「ルートで分ける」さらに高速な解法**

実は、この問題は $O(\sqrt{N})$ で解くことができます。一体どのようなことをすればこんな速い計算量でこの問題を解くことができるのでしょうか！？

まず、$A = floor(\sqrt{N})$ とします。そのとき、次の 3 パターンについて、条件を満たすような $(P_1, P_2, \ldots, P_{10}, Q_1, Q_2, \ldots, Q_{10})$ の組み合わせの個数を求めることを考えます。

パターン #1： $P_{max} \leq A$ である。
パターン #2： $Q_{max} \leq A$ である。
パターン #3： $P_{max} \leq A$ と $Q_{max} \leq A$ が両方成り立つ。

そこで、全体の組み合わせの個数は (パターン #1 を満たす組み合わせの個数) + (パターン #2 を満たす組み合わせの個数) − (パターン #3 を満たす組み合わせの個数) となります。パターン #3 の場合を引かなければならないのは、パターン #1, #2 に重複して数えあげられているからです。また、パターン #1, #2 を両方満たさない場合はなく、これは $A = floor(\sqrt{N})$ であることから分かります。

次に、パターン #1, #2, #3 を満たす場合の数を求めることを考えます。
まず、パターン #1 です。先ほどの $O(N)$ 解法と同じように、P_{max} を全探索すれば、計算量 $O(A)$ で場合の数が求まります。この場合の数を X とします。パターン #2 は、パターン #1 の P と Q を逆にしただけなので、場合の数はパターン #1 と全く同じです。
最後に、パターン #3 の場合の数はいくつでしょうか？$P_{max} \leq A, Q_{max} \leq A$ を満たすものはすべて $P_{max} \times Q_{max} \leq N$ となります！なぜなら、$A = floor(\sqrt{N})$ だからです。そうすると、パターン #3 を満たす場合の数は簡単に、「$P_{max} \leq A$ である $P_1, P_2, \ldots, P_{10}$ の組み合わせの数 A^{10} 通り」×「$Q_{max} \leq A$ である $Q_1, Q_2, \ldots, Q_{10}$ の組み合わせの数 A^{10} 通り」= A^{20} 通りとなります。

これより、答えは $2X - A^{20}$ 通りであることが分かります。先ほど述べた通り、X は計算量 $O(A) = O(\sqrt{N})$ で求められるので、この問題の答えも $O(\sqrt{N})$ で求めることができました！

サンプルコード (C 言語)： https://beta.atcoder.jp/contests/ddcc2019-qual/submissions/3636019

D. チップ・ストーリー　～黄金編～

さて、整数 N を P 進法で表したときの各位の数字の和が a_P のとき、N を $P-1$ で割った余りは何になるのでしょうか？

まず、P 進法で N を表したときの各桁を 1 の位から順に $b_0 b_1 b_2 b_3 \ldots b_M$ とします。$N = P^0 b_0 + P^1 b_1 + \cdots + P^M b_M$ です。そこで、N を $P-1$ で割った余りを考えます。

前提として、$P^0 \equiv P^1 \equiv P^2 \equiv \cdots \equiv P^M \equiv 1 \ (mod\ P-1)$ です。これは数学的には自明です。例えば $P = 10$ の場合、$1, 10, 100, 1000, \ldots$ を 9 で割った余りは全て 1 という事になります。 … ①

① より、$N = P^0 b_0 + P^1 b_1 + \cdots + P^M b_M \equiv 1 b_0 + 1 b_1 + \cdots + 1 b_M \equiv b_0 + b_1 + \cdots + b_M \ (\mathrm{mod}\ \mathrm{P}-1)$ となります。$b_0 + b_1 + b_2 + \cdots + b_M$ は各位の数字の和なので、**N を P で割った余りと各位の数字の和を $P-1$ で割った余りは同じと言えます。**

ですので、問題を次のように言い換えることができます。

整数 N を P 進法で表したときの各位の数字の和は a_P である。
↓　↓　↓
整数 N を $P-1$ で割った余りは a_P である。

この問題は、中国剰余定理（※注参照）を利用すれば解くことができます。また、これを使わなくても、枝刈り全探索を用いて解くことができます。

また、この問題には答えが最大でも 1 つしかありません。何故なら、$1, 2, 3, \ldots, 29$ の最小公倍数が 2329089562800 となり、10^{12} を超えるからです。

なお、一点注意が必要です。割った余りに言い換えた問題に 1 つの解があったとしても、実際に各位の数字の和の条件を満たすとは限らないことに注意してください。

サンプルコード（C++）： https://beta.atcoder.jp/contests/ddcc2019-qual/submissions/3633794

注： 中国剰余定理は https://qiita.com/drken/items/ae02240cd1f8edfc86fd を参照。