

bit全探索


kya(@tsaayk)

TMU-CS B4

June 26, 2022

bit全探索とは

- bit演算を用いて 2^n 通りの状態を全探索するテクニック
 - 「オン/オフ」, 「使う/使わない」, 「買う/買わない」などの 2 通りの選択肢が複数ある状況で有効
 - 2 通りの選択肢を 0/1 で表現する
- あり得る全ての状況を列挙して, それぞれの場合において計算や判定を行う

商品 3	商品 2	商品 1		商品 3	商品 2	商品 1
買う	買わない	買う		1	0	1

bit全探索とは

- あり得る全ての状況の列挙とは？
 - i -bit目が 0 なら買う, 1 なら買わない

商品 3	商品 2	商品 1		商品 1	商品 2	商品 3		10進数
買わない	買わない	買わない		0	0	0		0
買わない	買わない	買う		0	0	1		1
買わない	買う	買わない		0	1	0		2
買わない	買う	買う	→	0	1	1	→	3
買う	買わない	買わない		1	0	0		4
買う	買わない	買う		1	0	1		5
買う	買う	買わない		1	1	0		6
買う	買う	買う		1	1	1		7

bit演算

- bit全探索ではbit列を扱う
→ どうやって実装する？

- 計算機上で整数は全て 2 進数で計算されている

0 → 0...000 1 → 0...001 2 → 0...010 3 → 0...011
4 → 0...100 5 → 0...101 6 → 0...110 7 → 0...111

- さらに 2^n 通りの状態は $0, 1, \dots, 2^n$ (10進数) で表せる

bit演算

bit全探索で使用するbit演算

ビットシフト

```
n = 14  # 1110(2)
print(n << 1)
#> 28  (11100(2))
print(n >> 2)
#> 3   (11(2))
```

bit論理積

```
a = 14  # 1110(2)
b = 11  # 1011(2)
print(a & b)
#> 10  (1010(2))
```

bit演算

- i bit目の求め方
- $n \gg i$
 - n を i 個右シフト
 - n の i -bit目が 1 bit目に来る
- $n \& 1$
 - n と $0\cdots001$ のbit論理積
 - 1-bit 目が 1 の時に 1, 0 のときに 0 になる

i -bit目の取得

```
bit = 14 # 1110(2)
for i in range(4):
    print(bit >> i & 1)
#> 0
#> 1
#> 1
#> 1
```

例題：問題

問題文

N 個の整数 A_0, A_1, \dots, A_{N-1} と整数 W が与えられます.

A_0, A_1, \dots, A_{N-1} の中からいくつか選んで総和を W にすることはできますか？

制約

- $1 \leq N \leq 20$
- $0 \leq A_i \leq 10^9$
- $0 \leq W \leq 10^9$

例題：解法

- A_0, A_1, \dots, A_{N-1} のそれぞれは「使う/使わない」の 2 通り
- bit全探索でそれぞれを「使う/使わない」を列挙する
- 各状態で使うものの総和を計算して, W と一致するか判定する

例題：実装 (Python)

- 2^n 通りの状態は for 文を使うと楽に表せる
- 各状態において i -bit目が 0 か 1 かは $\text{bit} \gg i \ \& \ 1$ で求められる

```
ans = False
for bit in range(1 << n):
    s = 0 # a の部分和

    for i in range(n):
        # i-bit目が 1 かどうか
        if bit >> i & 1:
            s += a[i]

    # 部分和が w と一致するかどうか
    if s == w:
        ans = True
```