bit全探索

kya(@tsaayk)

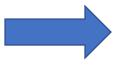
TMU-CS B4

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
買う	買わない	買う



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

商品 1	商品 2	商品 3	
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

	10進数
	0
	1
	2
	3
	4
	5
	6
	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

$$0 \rightarrow 0 \cdots 000$$
 $1 \rightarrow 0 \cdots 001$ $2 \rightarrow 0 \cdots 010$ $3 \rightarrow 0 \cdots 011$

$$4 \rightarrow 0 \cdots 100 \quad 5 \rightarrow 0 \cdots 101 \quad 6 \rightarrow 0 \cdots 110 \quad 7 \rightarrow 0 \cdots 111$$

• さらに 2^n 通りの状態は $0,1,\cdots,2^n$ (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)</pre>
#> 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 >> 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…001 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 $A_0, A_1, \cdots, A_{N-1}$ と整数 W が与えられます. $A_0, A_1, \cdots, A_{N-1}$ 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

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

例題:解法

 \bullet $A_0, A_1, \cdots, A_{N-1}$ 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 bit >> i & 1 で求めら れる

```
ans = False
for bit in range(1 << n):</pre>
   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
```