

HCPC新歓2022

## 問題解説 - Candies

解説 : tardigrade(@akTARDIGRADE13)

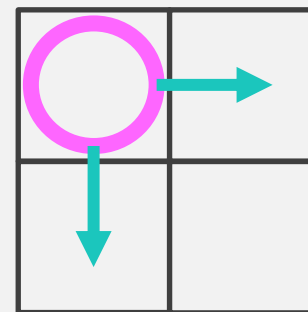
# 問題概要

- $2 \times N$ のマスが与えられる

3	2	2	4	1
1	2	2	2	1

- 初期位置は左上

- 右or下に移動することを繰り返す



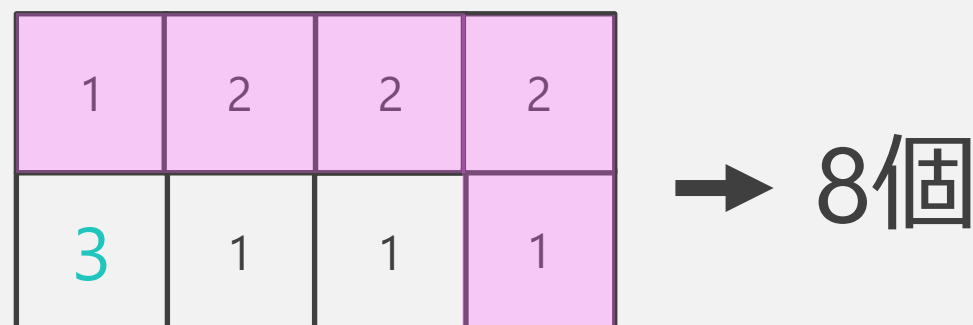
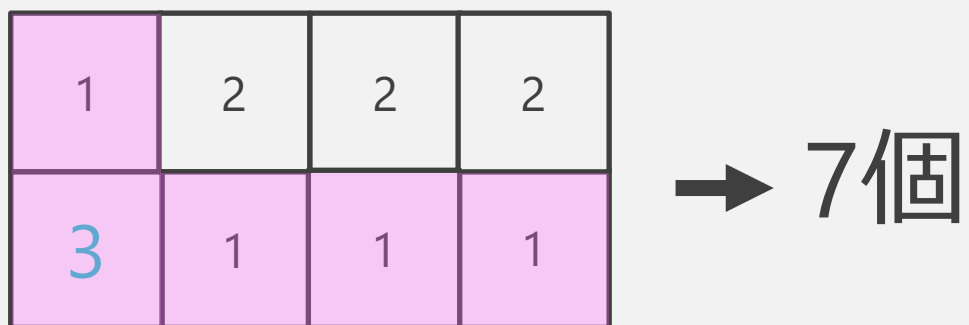
- 右下にたどり着くまでに一番アメを集められるルートを見つけ、集められた数を答えよ

3	2	2	4	1
1	2	2	2	1

→ 14個！

# 考察

- アメが一番多く置いてあるマスを通るルートを選ぶべき？
- 以下のようなケースが反例



- 何かしらの法則を見つけて最適ルートを選ぶのは難しそう

# 考察

- 全てのルートを試すことはできないか
- 2×5マスの場合

3	2	2	4	1
1	2	2	2	1

3	2	2	4	1
1	2	2	2	1

3	2	2	4	1
1	2	2	2	1

3	2	2	4	1
1	2	2	2	1

3	2	2	4	1
1	2	2	2	1

→ 5通り

# 考察

---

- $2 \times N$ マスならルートは $N$ 通り
- ルートが決まってる  $\rightarrow$  得られるアメの個数は $N+1$ 個のマスを見ればわかる
- 計算する回数は $N \times (N+1)$ 回
- 制約は $N \leq 100$ であるため、最大でも10000回くらいの計算で済む
- ルートを全部試すことが可能！！

# まとめ

- ルートは下に移動するタイミングで決まる → それを全て試す！
- その中で最もアメを集められたものが答え

ACコード (Python3) →

```
1. N = int(input())
2. A = list(map(int, input().split()))
3. B = list(map(int, input().split()))
4. ans = 0
5. for i in range(N):
6.     cnt = 0
7.     for j in range(i+1):
8.         cnt += A[j]
9.     for j in range(i, N):
10.        cnt += B[j]
11.    ans = max(ans, cnt)
12. print(ans)
```

# 余談

ピンク部分の総和と青部分の  
総和を求めたい

3	2	2	4	1
1	2	2	2	1

(ピンク)=(赤)+4  
(青)=(緑)-2      と計算できる

3	2	2	4	1
1	2	2	2	1

一個前の計算を利用できる  
毎回1から総和を求めなくてもよい

# 余談

- 適切に総和を管理するとループが1重で済む(高速化)
- 2重ループを回すコードの実行時間 255 ms (PyPy3)
- 1重ループを回すコードの実行時間 69 ms (PyPy3)

ACコード (Python3) →

```
1. N = int(input())
2. A = list(map(int, input().split()))
3. B = list(map(int, input().split()))
4. ans = 0
5. left = 0
6. right = sum(B)
7. for i in range(N):
8.     left += A[i]
9.     ans = max(ans, left+right)
10.    right -= B[i]
11. print(ans)
```



# 参考提出

---

- C++ (7 ms)

<https://atcoder.jp/contests/abc087/submissions/33876825>

- PyPy3 (255 ms)

<https://atcoder.jp/contests/abc087/submissions/33876645>

- C++ (6 ms) (高速化ver.)

<https://atcoder.jp/contests/abc087/submissions/33877476>

- PyPy3 (69 ms) (高速化ver.)

<https://atcoder.jp/contests/abc087/submissions/33877272>