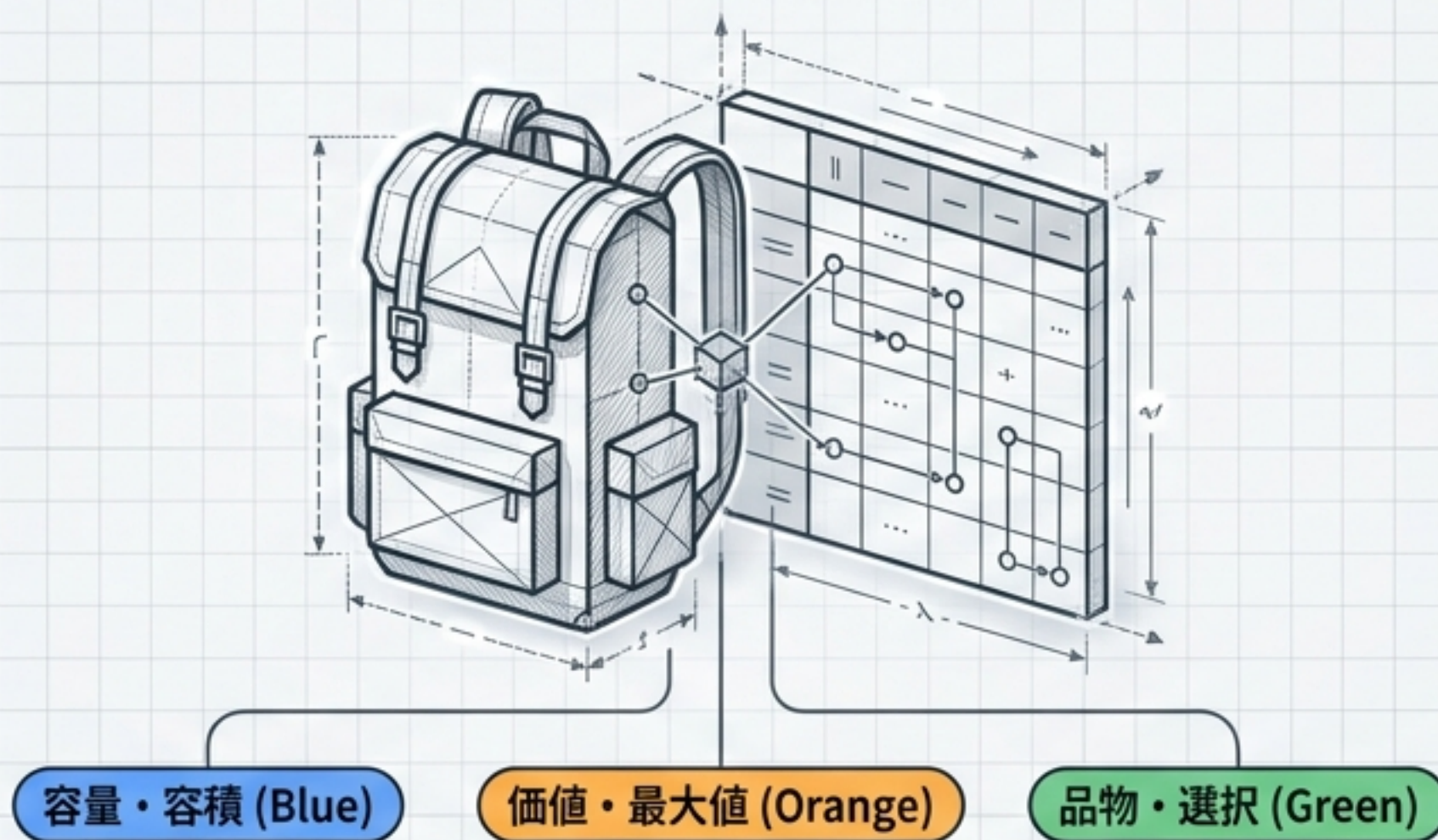


動的計画法を解説する：ナップザック問題 完全攻略ヒント集

平成29年度 秋期 応用情報技術者試験 午後 問3 (※解答のネタバレなし)



【台本】 応用情報技術者試験の午後問題、特にアルゴリズムと動的計画法 (DP) と聞くと、身構えてしまう方も多いかもしれません。でも大丈夫です。このスライドは、答えを丸暗記するためのものではありません。あなた自身の手でパズルを解き明かすための「ヒント集」です。コードと表の関係性を一緒に解説していきましょう。

全探索（すべての組み合わせを試す）

無数の可能性

計算爆発

重複計算

複雑すぎる

動的計画法（過去のベストアンサーを再利用して表を埋める）

	0	4	7	10	...	22
0	15					
4						
⋮						
					15 + ...?	

- 小さな問題から順番に解く。
- 過去の計算結果を表にメモし、再計算を防ぐ。

【台本】動的計画法と聞くと難しそうですが、本質はとてもシンプルです。全ての組み合わせを闇雲に試すのではなく、「小さなナップザック」の最適な答えを先に計算して表にメモしておき、少し大きなナップザックを考える時にそのメモを再利用する。つまり、決められたルールに従って「表を埋めるだけのゲーム」なのです。

黄金のルール (The Golden Rule)

1. 上のマスをもそのまま引き継ぐか？（追加しない）
2. 容量を「品物の容積」分だけ戻り、「品物の価値」を足すか？（追加する）

勝負して、大きい方を採用！

表2 選んだ品物の最大の価値合計（作成途中）

	容量制限					
	0	1	2	3	4	5
A だけを選ぶ	0	2	4	6	8	10
A, B を選ぶ	0	2		8	12	14

価値を足す（価値 6）

VS

容量を戻る（容積 2）

【台本】表を埋めるための「黄金のルール」を確認しましょう。新しい品物を入れるかどうか迷ったら、2つの選択肢を比較します。1つ目は「入れない」場合。これは単純に一つ上のマスの価値を引き継ぎます。2つ目は「入れる」場合。ここが重要です。品物の「容積」の分だけ表を左に戻り、そこに品物の「価値」を足します。この2つを比べて、大きい方がそのマスの勝者となります。

表4 品物A, B, Cを選ぶ場合の最大の価値合計

	容量制限					
	0	1	2	3	4	5
Aだけ	0	2	4	6	8	9
A, Bを	0	2	?	?	9	12
A, B, Cを選	0	2	? [ア]	? [イ]	12	14

Cheat Sheet

品物 C

容積 = 3

価値 = 9

品物Cの容積は「3」。容量制限が「2」のナップザックに、そもそもCは入るのか？入らない場合、黄金のルールのどちらを選ぶ？

【台本】では設問1の[ア]と[イ]を考えてみましょう。新しく品物Cが追加されました。品物Cの容積は「3」です。ここで[ア]のマス（容量制限2）に注目してください。容量が2しかないカバンに、容積3の品物は物理的に入るでしょうか？入らない場合、先ほどのルールの「入れない」選択肢しか選べませんよね。

表4 品物 A, B, C を選べる場合の最大の価値合計

Cheat Sheet

品物 C
容積 = 3 [Blue]
価値 = 9 [Orange]

容量	0	1	2	3	4	5
A, B を選べる	0	2	6	8	12	14
A, B, C を選べる	0	2	?	?	?[ウ]	?[エ]

Path 1
そのまま引き継ぐ
(Value 12)

Path 2
2 + 9 = ?

VS

+9

容量4と5では、品物Cが入るスペースがある！
「Cを入れない場合の価値」と「Cを入れた場合の価値」、どちらが大きいかな勝負させよう。

【台本】

【台本】 続いて[ウ]と[エ]です。容量が4や5になれば、ついに品物Cを入れるスペースが生まれます！
ここで黄金のルールの出番です。すぐ上のマスの価値と、品物Cの容積「3」だけ左に戻って品物Cの価値「9」を足した数字。
この2つを計算して「VS（バーサス）」、つまり比較をして、より大きな数字をマスに書き込んでください。

最大価値を生み出した「中身」はどうやって特定する？
ゴールから出発し、選んだ品物の容積分だけ「逆戻り」して足跡をたどろう。

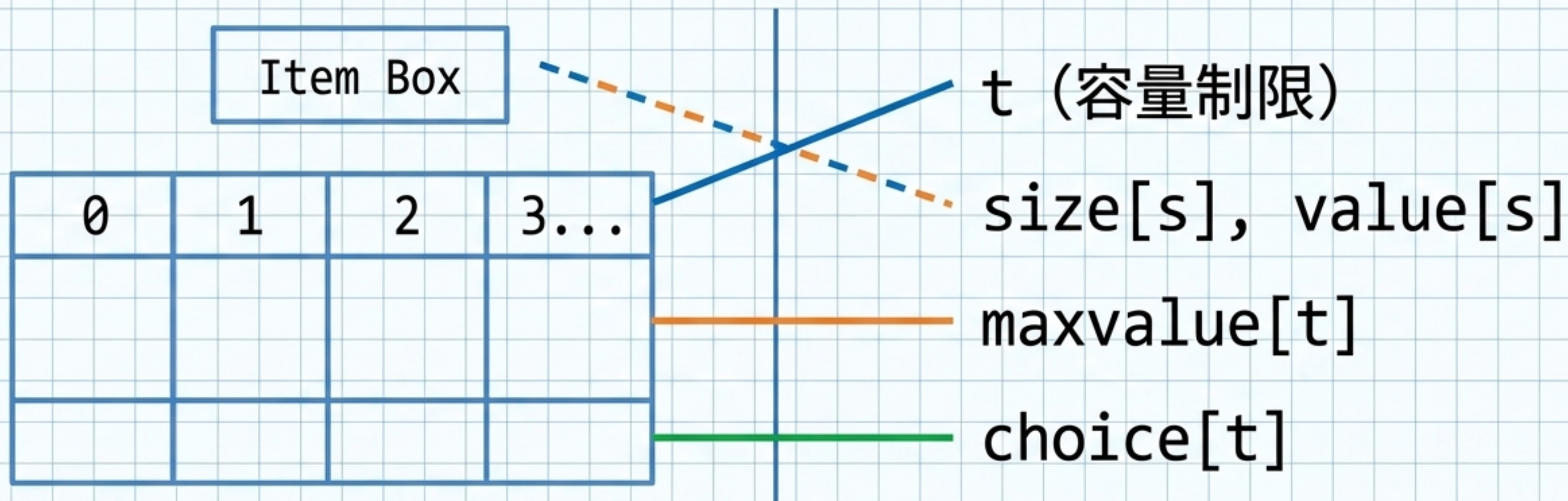
表3 最後に選んだ品物

容量制限	0	1	2	3	4	5
品物	なし	A	B	B	B	B



【台本】設問1の(2)は、最大価値の中身（どの品物を選んだか）を答える問題です。これは「逆戻りの足跡探し」です。ゴールである容量5のマスを見ると、最後に選んだのは「B」だと分かります。Bを入れたということは、カバンの容量を「2」使ったということ。つまり、次は5引く2で、容量3のマスを見に行きます。これを容量がゼロになるまで繰り返すだけです。

プログラムは、私たちが手作業でやった「表埋め」を自動化しているだけ。変数の役割を色で紐解こう。



【台本】ここから設問2、いよいよプログラムの解説です。コードを英語の文章のように読むはいけません。プログラムは、私たちが今手作業でやった「表埋めゲーム」を自動で行っているだけです。表の列の数字が「t」、書き込んだ価値が「maxvalue」、最後に選んだ品物が「choice」です。この対応関係を頭に入れておきましょう。

Algorithm Blueprint



temp $\leftarrow$ maxvalue[**[オ]**] + value[s]
if (temp **が** maxvalue[t] **より大きい**)
 [力] $\leftarrow$ temp

容量tから、品物sの容積を引くには？



勝負に勝ったら、現在の容量tの最高記録を更新する！

【台本】空欄[オ]と[力]のヒントです。[オ]を含む行は「品物を入れた場合の価値」を計算しています。表でやった「品物の容積分だけ左に戻る」という動きを数式にするには、現在の容量「t」から、品物の容積をどうすればよいでしょうか？
そして[力]は比較の後の処理です。新しい価値「temp」のほうが大きかった場合、その容量「t」の最高記録を上書き保存する必要があります。どこに保存しますか？

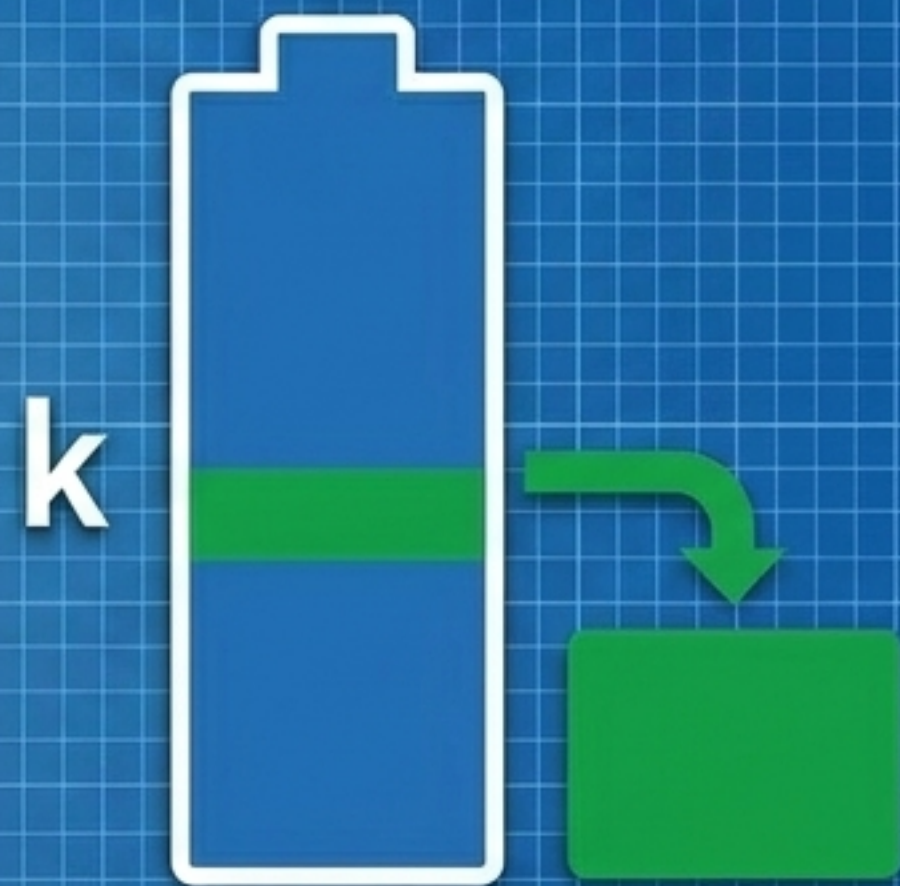
Algorithm Blueprint

ループの中で品物を出力したら、残りの容量 k を減らす。
今まさに出力した「その品物」の容積を引くには？

```
while (choice[k] が 0 以上)
```

```
    choice[k] を出力
```

```
    k ← k - size[ キ ]
```



【台本】空欄[キ]は、先ほどの「逆戻りの足跡探し」をコードにした部分です。変数「 k 」は現在の残り容量を表しています。品物を一つ出力したら、その品物が占めていたスペース分だけ、残り容量「 k 」から引き算しなければなりません。すぐ上の行で「choice[k]」を出力しましたね。では、その「たった今選んだ品物」の容積を呼び出すには、インデックスに何を指定すべきでしょうか？

[ク] を出力 // 価値合計を出力する。

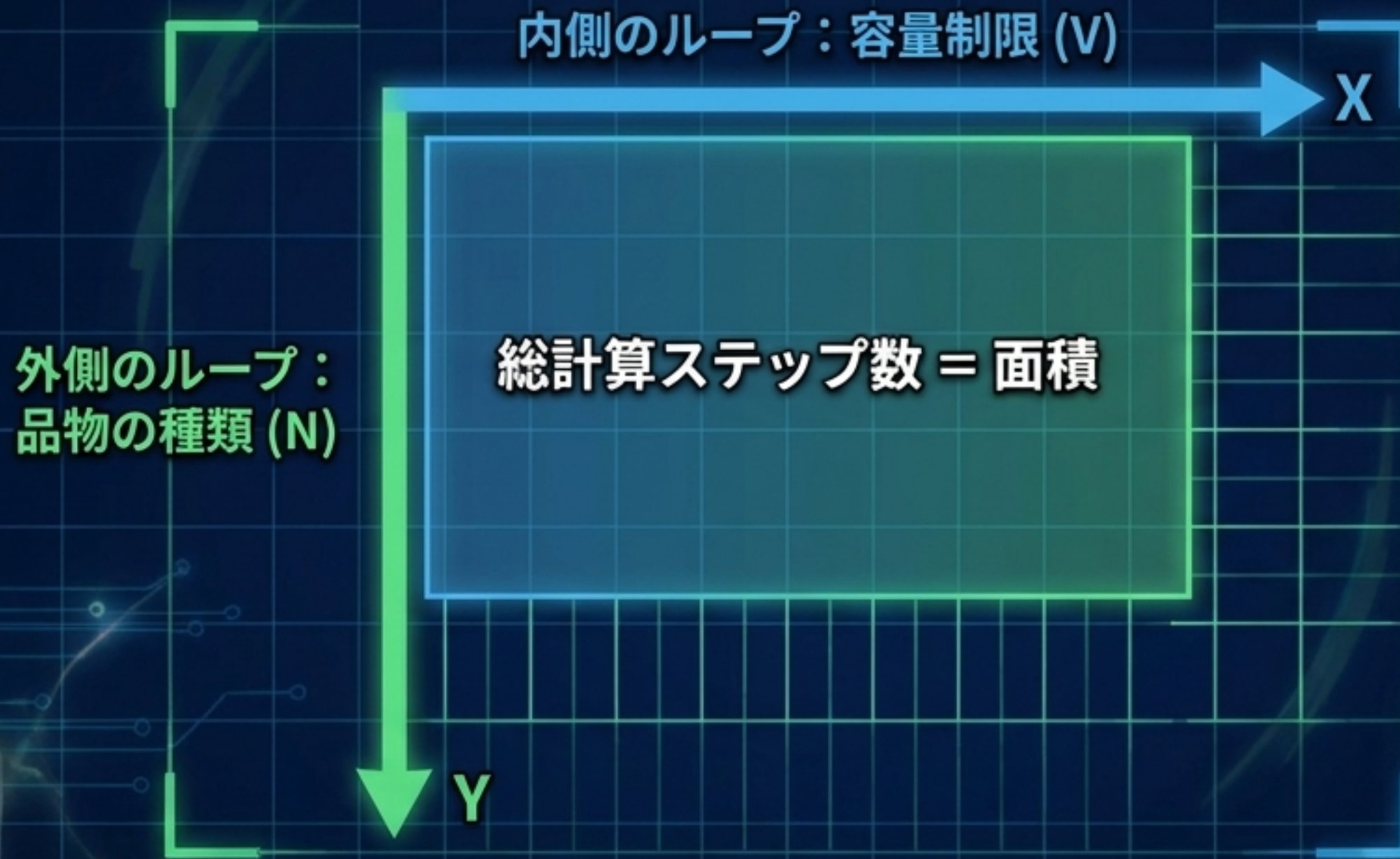


全てのループが終わった後、ナップザック全体の「最終的な最大価値」は、
表のどこのマスに保存されている？

【台本】コードの最後、空欄[ク]です。すべての計算が終わった後、最終的な価値の合計を出力します。私たちが表を埋めたとき、最終的な答えはどこにありましたか？そう、表の一番右端です。では、価値の最高記録を保存している配列「maxvalue」の中で、カバンの総容量「V」に対応する一番右端のマスプログラムで表現するとどうなるでしょうか？

Algorithm Blueprint

計算量のオーダー $O(\quad)$ は、ループが合計で何回まわるか？



【台本】最後の設問、空欄[ケ]の計算量（オーダー）についてです。計算量は、プログラム内のループが合計で何回実行されるかに比例します。外側のループは品物の種類「N」の数だけ回ります。そして、その1回1回の中で、内側のループが容量制限「V」の数だけ回ります。縦がN、横がVの長方形の面積を求めるように、この2つの変数をどう組み合わせれば総ステップ数になるか考えてみてください。

Algorithm Blueprint

	1	2	3	4
0		2	3	4
1	2	1	5	4
2	1	3	4	3
3	2	2	2	

DPは「表埋めゲーム」。迷ったら黄金のルール！



組み合わせの特定は、ゴールからの「逆戻り」！



コードは読まずに、表の動きと「翻訳」する！



【台本】お疲れ様でした！動的計画法のアルゴリズムは、一見複雑な数式に見えますが、実は私たちが手作業でできる「表埋め」を忠実に再現しているだけです。本番の試験でも、コードだけで悩まず、まずは余白に小さな表を書いて手を動かしてみてください。コードの意味が自然と浮かび上がってくるはずです。あなたの合格を応援しています！