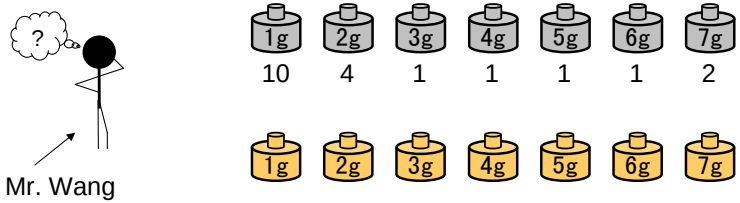
Optimal Binary Search Tree

Speaker: Chung-Chin Kuo
Produced by Jsing Lin

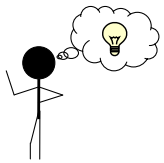
Mr. Wang's problem

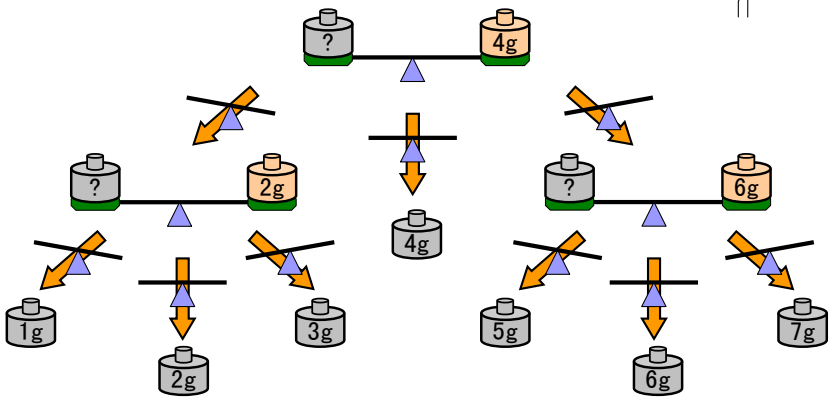
- There are many weights of 1g ~ 7g in the laboratory.
- Mr. Wang has a set of standard weight of 1g ~ 7g.
- He needs to classify those weights.



Mr. Wang's solution

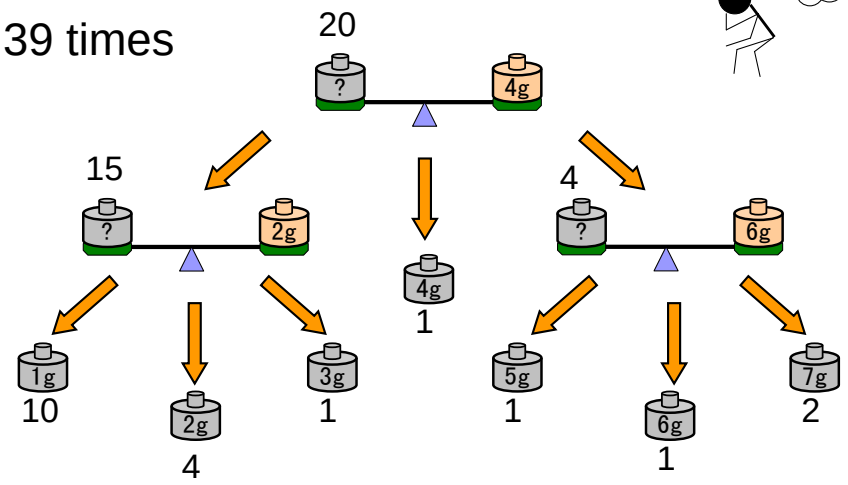
- Using a balanced search tree.

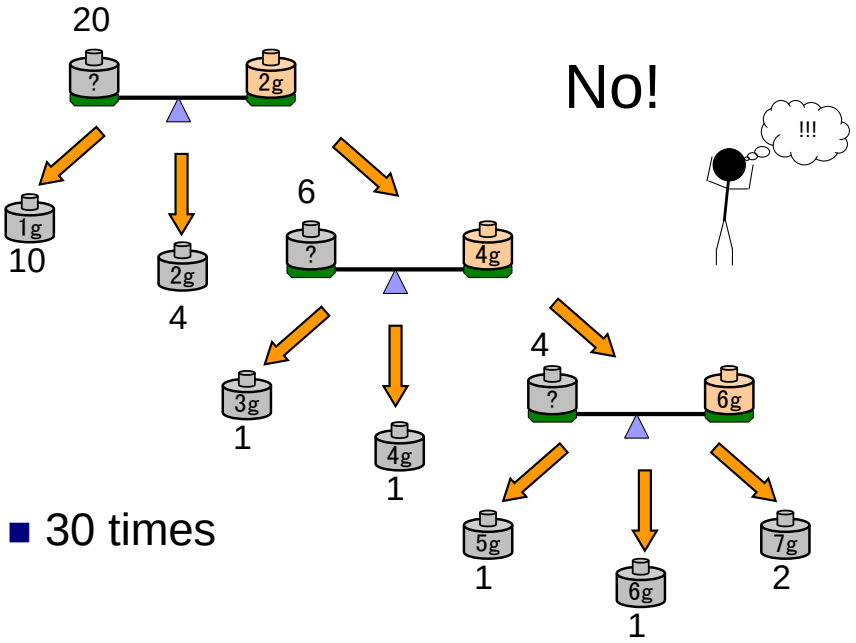




Is it optimal?

- 39 times



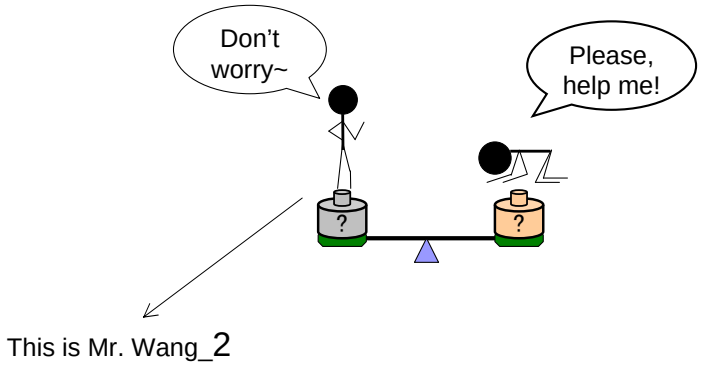


No!

- 30 times

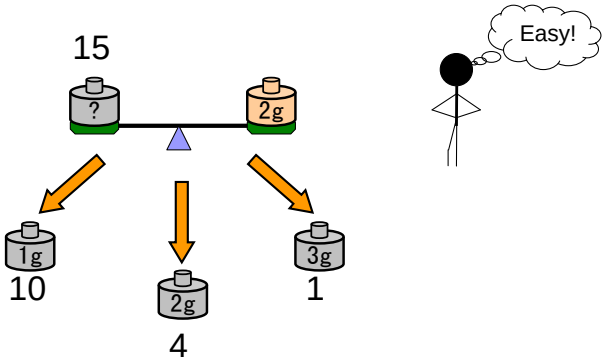
Please help Mr. Wang!

- What is the optimal solution?



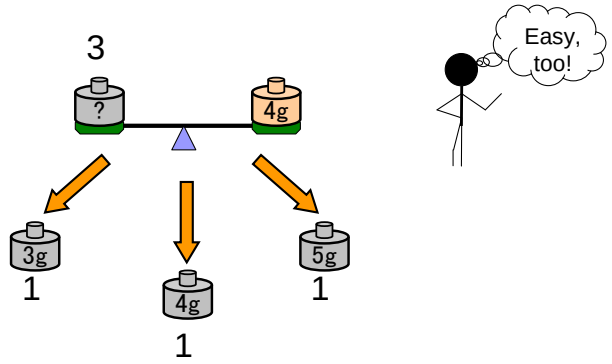
What if…

- If there are only weights of 1g, 2g, and 3g.
- 15 times.



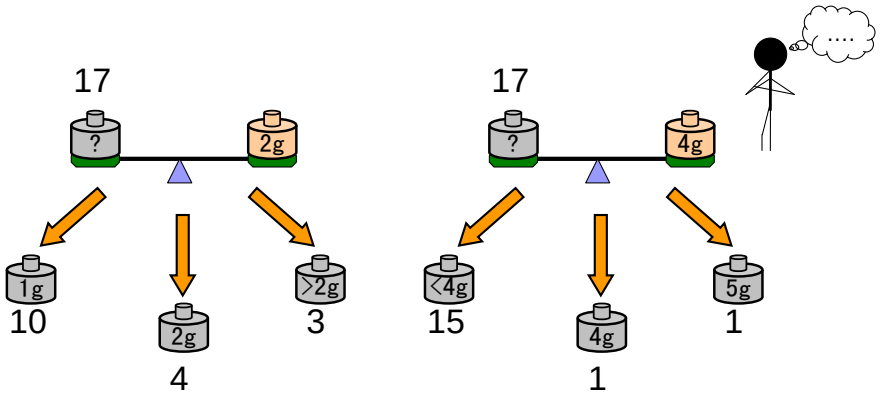
What if…

- If there are only weights of 3g, 4g, and 5g.
- 3 times.

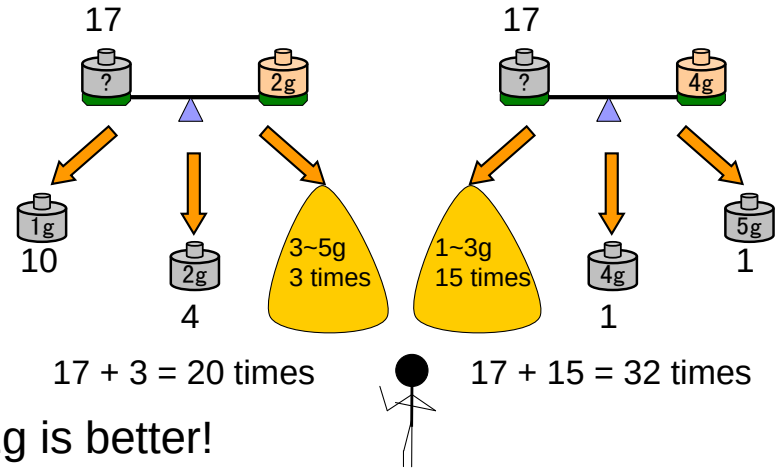


What if…

- What if there are weights of 1g ~ 5g?



Which one is better?

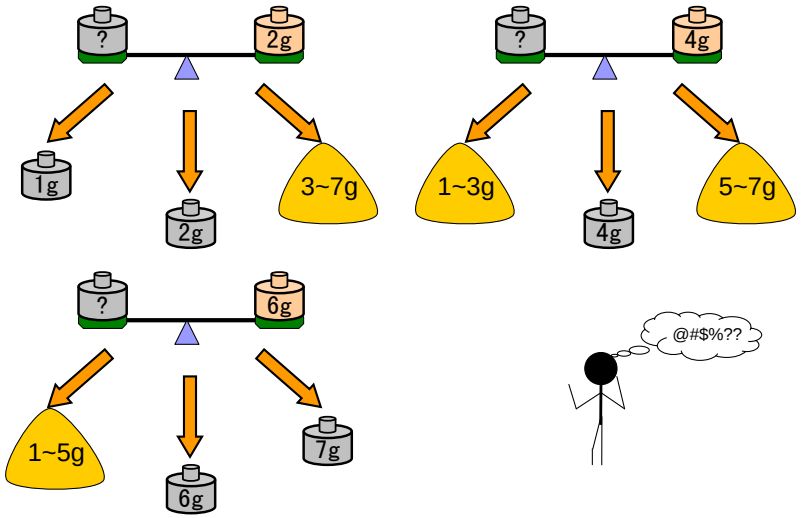


17 + 3 = 20 times

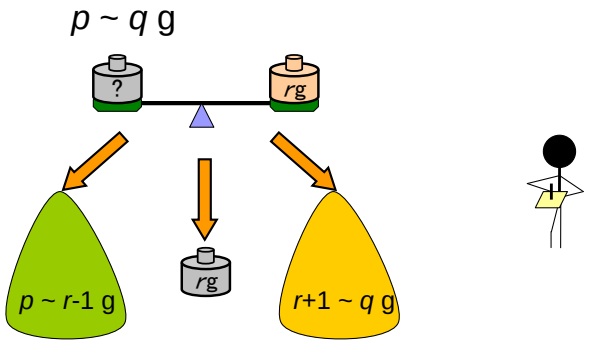
17 + 15 = 32 times

- 2g is better!

Which one is better?



Let's consider general problems



- time: ▲ + ▲ + number of $p \sim q$ g
- ▲ and ▲ must be minimal!

What is optimal?

- Select *r* such that  is minimal.
- Let *mtime*(*p*, *q*) be the minimal comparison times for weights of *p* ~ *q* g.
  - *number*(*p*, *q*) is the number of weights in *p* ~ *q* g
- *mtime*(*p*, *q*) = number(p, q) + $\min_{r=p+1 \text{ to } q-1}\{mtime(p, r-1) + mtime(r+1, q)\}$
  - if $p < q$
- *Mtime(p, q)* = 0
  - if $p = q$

Dynamic programming

number(*p*, *q*)

	1	2	3	4	5	6	7
1	10	14	15	16	17	18	20
2		4	5	6	7	8	10
3			1	2	3	4	6
4				1	2	3	5
5					1	2	4
6						1	3
7							2

prefix sum:
Let *pre*(*p*) be the number of 1 to p.
number(*p*, *q*) = *pre*(*q*) – *pre*(*p* – 1)

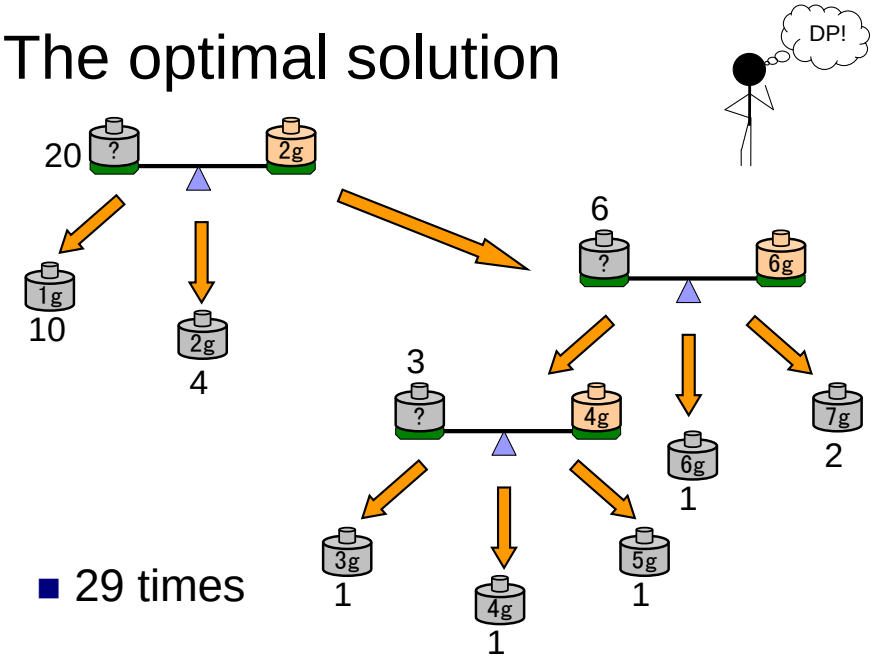
Dynamic programming

mtime(*p*, *q*)

	1	2	3	4	5	6	7
1	0	14	15	18	20	24	29
2		0	5	6	9	11	17
3			0	2	3	6	9
4				0	2	3	7
5					0	2	4
6						0	3
7							0

Time: $O(n^3)$
Space: $O(n^2)$

The optimal solution



- 29 times