01-C3

绪论

渐近复杂度：指数

慌得那拿盘的小怪，战兢兢跑去报道：“难，难，难！难，难，难！”
老妖道：“怎么有许多难？”

“你是什么东西？”太太说。四虎子也楞住了，他自己不知道他是什么东西——这本是世上最难答的一个问题。

邓俊辉
deng@tsinghua.edu.cn

$\mathcal{O}(2^n)$: exponential

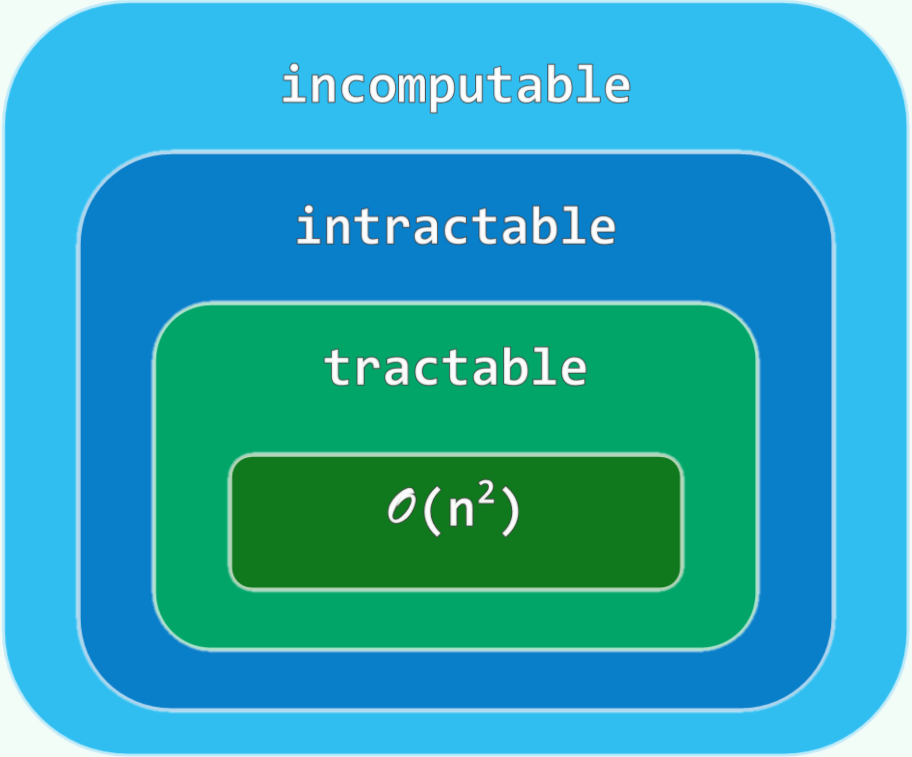
- ❖ **指数:** $T(n) = \mathcal{O}(a^n), \quad a > 1$

$$\because \; e^n \; = \; 1 + n + n^2/2! + n^3/3! + n^4/4! + \ldots$$

$$\therefore \forall \; c > 1, \; n^c = \mathcal{O}(2^n)$$

$$n^{1000...01} = \mathcal{O}(1.000\ldots01^n) = \mathcal{O}(2^n)$$

$$1.000\ldots01^n = \Omega(n^{1000...01})$$



incomputable

intractable

tractable

$\mathcal{O}(n^2)$

- ❖ **这类算法的计算成本增长极快，通常被认为不可忍受**
- ❖ **从$\mathcal{O}(n^c)$到$\mathcal{O}(2^n)$，是从有效算法到无效算法的分水岭**
- ❖ **$\mathcal{O}(2^n)$算法往往显而易见，然而设计出$\mathcal{O}(n^c)$算法却极其不易，有时甚至注定是徒劳无功**
- ❖ **更糟糕的是，这类问题要远比我们想象的多得多...**

SubsetSum：问题

$$\forall\ S = \{a_1, a_2, \ldots, a_n\} \subset \mathbb{Z}^+ \quad \text{and} \quad 0 \le M \le \sum_{k=1}^{n} a_k$$

$$\exists\ T \subset S \ \text{ s.t. } \sum\nolimits_{a\in T} a = M \ \ ?$$

❖ **选举人团投票制：**

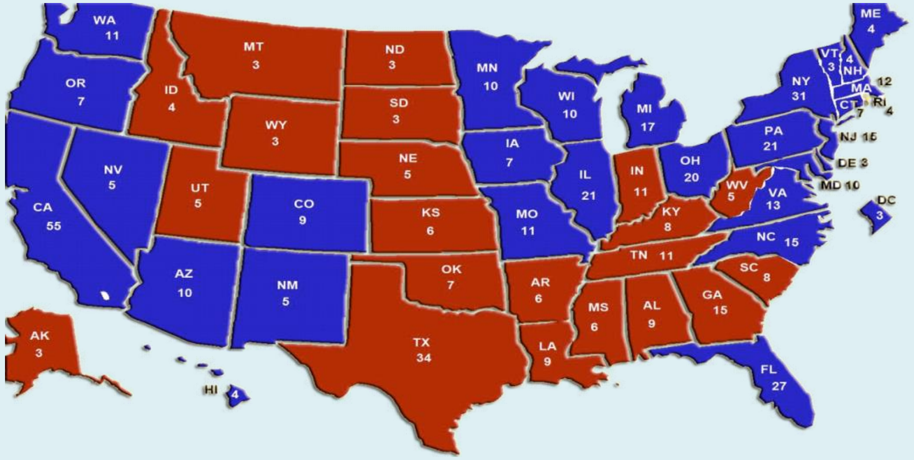
50个州加1个特区，共538票 //n, 2M

获270张选举人票，即可当选 //M+1

❖ **若仅两位候选人，会否恰好各得269票？** //M

❖ **可视作SubsetSum的特例：** $\displaystyle\sum_{k=1}^{n} a_k = 2M$

55	California	11	Indiana	7	Connecticut	4	Idaho
34	Texas	11	Missouri	7	Iowa	4	Maine
31	New York	11	Tennessee	7	Oklahoma	4	New Hampshire
27	Florida	11	Washington	7	Oregon	4	Rhode Island
21	Illinois	10	Arizona	6	Arkansas	3	Alaska
21	Pennsylvania	10	Maryland	6	Kansas	3	Delaware
20	Ohio	10	Minnesota	6	Mississippi	3	D. C.
17	Michigan	10	Wisconsin	5	Nebraska	3	Montana
15	Georgia	9	Alabama	5	Nevada	3	North Dakota
15	New Jersey	9	Colorado	5	New Mexico	3	South Dakota
15	North Carolina	9	Louisiana	5	Utah	3	Vermont
13	Virginia	8	Kentucky	5	West Virginia	3	Wyoming
12	Massachusetts	8	South Carolina	4	Hawaii	538 = ∑	



SubsetSum: ~~算法~~ 程序

❖ 直觉上，并不难：

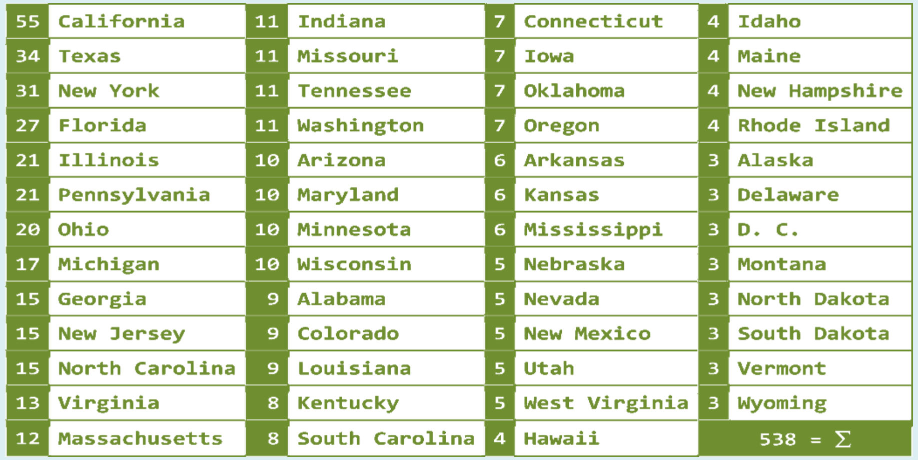
逐一枚举S的每一子集，统计总和并核对

❖ sSum(S = {a_1, a_2, ..., a_n}, M)

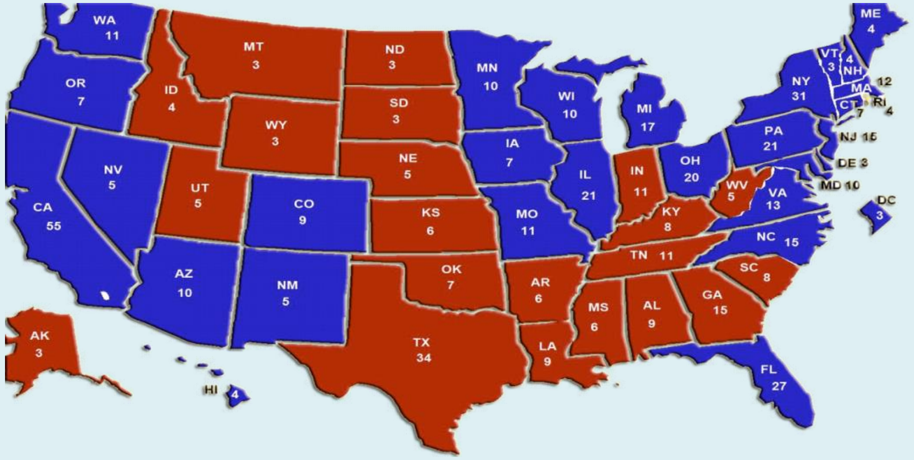
```
    if ( M == 0 ) return true;
    if ( n == 0 ) return false;
    S = S\{a_n}; //classification
    return sSum(S, M) || sSum(S, M-a_n);
```

❖ 最坏情况下，需要检视每一个子集，然而...

$$|2^S| = 2^{|S|} = 2^n$$



55	California	11	Indiana	7	Connecticut	4	Idaho
34	Texas	11	Missouri	7	Iowa	4	Maine
31	New York	11	Tennessee	7	Oklahoma	4	New Hampshire
27	Florida	11	Washington	7	Oregon	4	Rhode Island
21	Illinois	10	Arizona	6	Arkansas	3	Alaska
21	Pennsylvania	10	Maryland	6	Kansas	3	Delaware
20	Ohio	10	Minnesota	6	Mississippi	3	D. C.
17	Michigan	10	Wisconsin	5	Nebraska	3	Montana
15	Georgia	9	Alabama	5	Nevada	3	North Dakota
15	New Jersey	9	Colorado	5	New Mexico	3	South Dakota
15	North Carolina	9	Louisiana	5	Utah	3	Vermont
13	Virginia	8	Kentucky	5	West Virginia	3	Wyoming
12	Massachusetts	8	South Carolina	4	Hawaii		538 = ∑



SubsetSum: NPC

- 故严格地讲，这类方法只能算作程序，而非算法

- 还是直觉：应该有更好的办法吧?

  比如...转化为...背包问题...?

  很遗憾，这里对于整数的取值范围未作任何假定

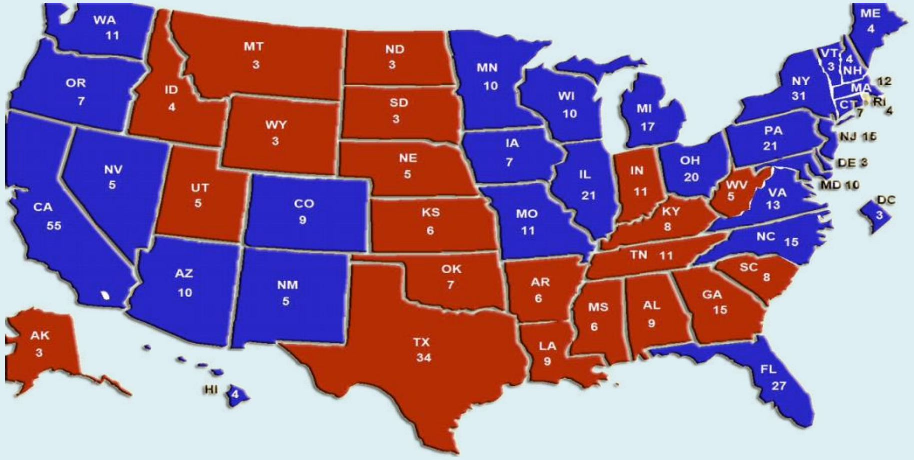
- 定理：SubsetSum is NP-complete —— 什么意思?

- 意即：就目前的计算模型而言

  不存在可在多项式时间内解决此问题的算法

  上述的直觉算法，居然就是最优的

55	California	11	Indiana	7	Connecticut	4	Idaho
34	Texas	11	Missouri	7	Iowa	4	Maine
31	New York	11	Tennessee	7	Oklahoma	4	New Hampshire
27	Florida	11	Washington	7	Oregon	4	Rhode Island
21	Illinois	10	Arizona	6	Arkansas	3	Alaska
21	Pennsylvania	10	Maryland	6	Kansas	3	Delaware
20	Ohio	10	Minnesota	6	Mississippi	3	D. C.
17	Michigan	10	Wisconsin	5	Nebraska	3	Montana
15	Georgia	9	Alabama	5	Nevada	3	North Dakota
15	New Jersey	9	Colorado	5	New Mexico	3	South Dakota
15	North Carolina	9	Louisiana	5	Utah	3	Vermont
13	Virginia	8	Kentucky	5	West Virginia	3	Wyoming
12	Massachusetts	8	South Carolina	4	Hawaii	538 = Σ	


WA 11
OR 7
CA 55
NV 5
ID 4
MT 3
WY 3
UT 5
AZ 10
CO 9
NM 5
ND 3
SD 3
NE 5
KS 6
OK 7
TX 34
MN 10
IA 7
MO 11
AR 6
LA 9
WI 10
IL 21
MI 17
IN 11
OH 20
KY 8
TN 11
MS 6
AL 9
GA 15
FL 27
SC 8
NC 15
VA 13
WV 5
PA 21
NY 31
ME 4
VT 3
NH 4
MA 12
RI 4
CT 7
NJ 15
DE 3
MD 10
DC 3
AK 3
HI 4