

Dynamic Programming

Backtracking



Memoize (remember past work)



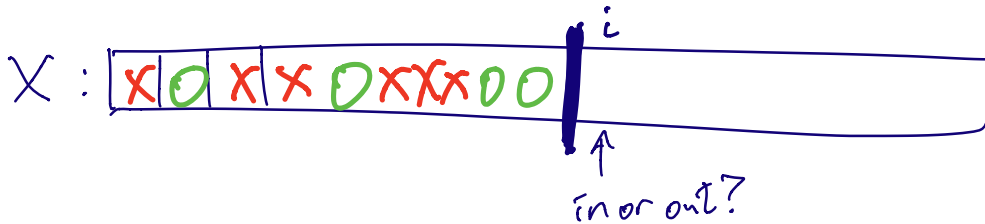
DP

recursive calls
are now table lookups

Subset Sum

array $X[1..n]$
Given set X of positive ints
 T positive int

Does any subset of X sum to T ?



$SS(i, t) = \text{TRUE}$ iff some subset of $X[i..n]$ sums to t
 FALSE otherwise

We want
 $SS(1, T)$

$$SS(i, t) = \begin{cases} \text{FALSE} & \text{if } i > n \text{ and } t > 0 \\ \text{TRUE} & \text{if } t = 0 \\ SS(i+1, t) & \text{if } t < X[i] \\ SS(i+1, t - X[i]) \vee SS(i+1, t) & \text{otherwise} \end{cases}$$

We want

$\Rightarrow O(2^n)$ time

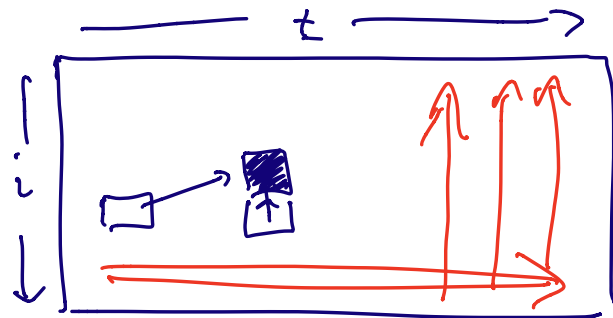
- Subproblems $1 \leq i \leq n+1$ and $0 \leq t \leq T$

- Memoize $S[1..n+1, 0..T]$ Σ^d array

- Dependencies

- Order

- Time = $O(nT)$

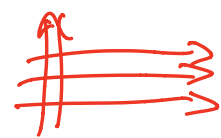


\Rightarrow outer loop
 \rightarrow inner loop

```

SUBSETSUM( $X[1..n], T$ ):
   $S[n+1, 0] \leftarrow \text{TRUE}$ 
  for  $t \leftarrow 1$  to  $T$ 
     $S[n+1, t] \leftarrow \text{FALSE}$ 

  for  $i \leftarrow n$  downto 1
     $S[i, 0] = \text{TRUE}$ 
    for  $t \leftarrow 1$  to  $X[i] - 1$ 
       $S[i, t] \leftarrow S[i+1, t]$      $\langle\langle \text{Avoid the case } t < 0 \rangle\rangle$ 
    for  $t \leftarrow X[i]$  to  $T$ 
       $S[i, t] \leftarrow S[i+1, t] \vee S[i+1, t - X[i]]$ 
  return  $S[1, T]$ 
  
```



Edit distance

Levenshtein dist.
Ulam dist.

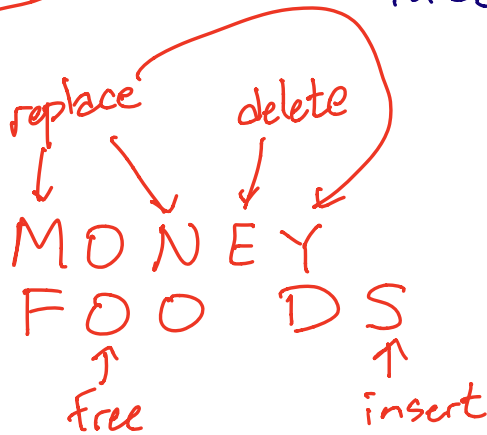
MONEY → FOOD

↓
MONED
↓
MOND
↓
MOOD
↓
FOOD

Min # operations

- replace
- insert
- delete

to change one string into another.



ALGORITHM ?
ALTRUISTIC .

ALGORITHM | M
ALTRUISTI | C

ALGOR | ITHM
ALTRUIST | I C

ALGORITHM |
ALTRUISTI | C

Input: A[1..m]
B[1..n]

ALGORITHM | M
ALTRUISTIC |

Edit(i,j) = edit distance
between A[1..i] and B[1..j]

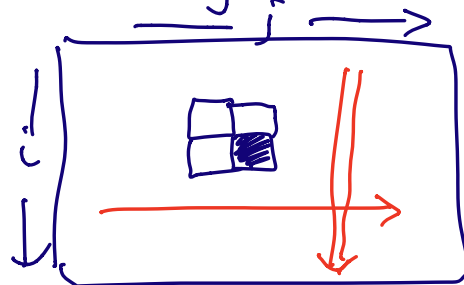
$$\text{Edit}(i,j) = \begin{cases} j & \text{if } i=0 \\ i & \text{if } j=0 \\ \min \begin{cases} \text{Edit}(i-1, j-1) + [A[i] \neq B[j]] & \text{rep} \\ \text{Edit}(i, j-1) + 1 & \text{ins} \\ \text{Edit}(i-1, j) + 1 & \text{del} \end{cases} \end{cases}$$

we want Edit(m,n)

- Subproblems
- Memoize
- Dependencies
- Order
- time $O(mn)$

$$0 \leq i \leq m$$

$$0 \leq j \leq n$$



$$\text{Edit}(i, j) = \begin{cases} i & \text{if } j = 0 \\ j & \text{if } i = 0 \\ \min \left\{ \begin{array}{l} \text{Edit}(i-1, j) + 1, \\ \text{Edit}(i, j-1) + 1, \\ \text{Edit}(i-1, j-1) + [A[i] \neq B[j]] \end{array} \right\} & \text{otherwise} \end{cases}$$

EDITDISTANCE(A[1..m], B[1..n]):

```

for j ← 1 to n
    Edit[0, j] ← j
for i ← 1 to m
    Edit[i, 0] ← i
    for j ← 1 to n
        if A[i] = B[j]
            Edit[i, j] ← min {Edit[i-1, j] + 1, Edit[i, j-1] + 1, Edit[i-1, j-1]}
        else
            Edit[i, j] ← min {Edit[i-1, j] + 1, Edit[i, j-1] + 1, Edit[i-1, j-1] + 1}
return Edit[m, n]

```

	A	L	G	O	R	I	T	H	M		
	0	→1	→2	→3	→4	→5	→6	→7	→8	→9	
A	↓ 1	↘ 0	→1	→2	→3	→4	→5	→6	→7	→8	
L	↓ 2	↓ 1	↘ 0	→1	→2	→3	→4	→5	→6	→7	
T	↓ 3	↓ 2	↓ 1	↘ 1	→2	→3	→4	↘ 4	→5	→6	
R	↓ 4	↓ 3	↓ 2	↘ 2	↘ 2	↘ 2	↘ 2	↘ 3	→4	→5	→6
U	↓ 5	↓ 4	↓ 3	↘ 3	↘ 3	↘ 3	↘ 3	↘ 3	→4	→5	→6
I	↓ 6	↓ 5	↓ 4	↘ 4	↘ 4	↘ 4	↘ 4	↘ 3	→4	→5	→6
S	↓ 7	↓ 6	↓ 5	↘ 5	↘ 5	↘ 5	↘ 5	↘ 4	↘ 4	↘ 5	↘ 6
T	↓ 8	↓ 7	↓ 6	↘ 6	↘ 6	↘ 6	↘ 6	↘ 5	↘ 4	→5	→6
I	↓ 9	↓ 8	↓ 7	↘ 7	↘ 7	↘ 7	↘ 7	↘ 6	↘ 5	↘ 5	→6
C	↓ 10	↓ 9	↓ 8	↘ 8	↘ 8	↘ 8	↘ 8	↘ 7	↘ 6	↘ 6	↘ 6

A L G O R I T H M
A L T R U I S T I C