More Dynamic Programming (DP)

Tuesday, March 10, 2026 9:20 AM

$L = \{0^i 1^i \mid i \geq 0\}$

$\bar{L} = \{1^i 0^i \mid i \geq 0\}$! NO!

$\bar{L} = \{x \in \{0,1\}^* \mid x \notin L\}$

	Due
HW6	today
HW7	Mar 24
GPS 7	Mar 23

DP:

1. Subproblem Def.
2. Recursive Formula
3. Evaluation Order

✡ Word Splitting:

Given a sequence of alphabets: $A = a_1 a_2 \ldots a_n$

Check if it can be split into valid words.

Given: IsWord(string) = TRUE if string is a valid word
= FALSE.

→ TRUE or FALSE

eg. thereisoneverywordsplitting → TRUE.

i → n

① Subproblem Def^n: $i = 1 \ldots n \ (n+1)$

split(i) = TRUE if $a_i \ldots a_n$ can be split into valid words

= FALSE o.w.

where split(n+1) we mean split(ε) → empty string.

② Recursive Formula:

Base case: Split(n+1) ($\equiv$ split(ε)) = TRUE.

$$\text{Split}(i) = \bigvee_{i \le j \le n} \left(\text{IsWord}(A[i,\ldots,j]) \wedge \text{Split}(j+1) \right)$$

split(i)

split(i)

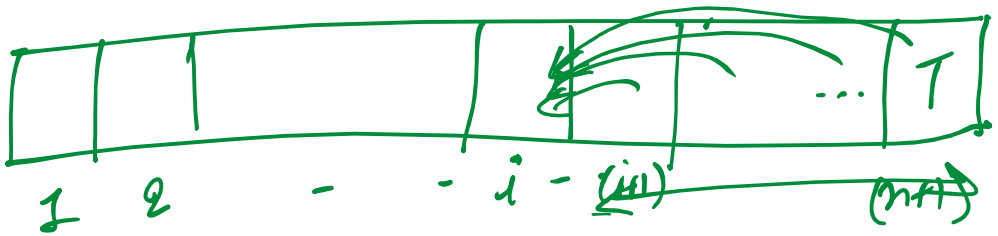
$a_i\ a_{(i+1)} \ldots a_n$

$i \le j \le n$

$S[i] = \text{split}(i)$

$S[(n+1)] = \text{TRUE}$

Ans: Split (1)



(3) Evaluation Order:

Decreasing order of i.

Time: $(n+1) * O(n) = O(n^2)$

subprob. ↑ time to solve each subprob.

Polynomial-time $\equiv O(\text{poly}(\text{i/p size})) = O(n^c)$

where c is a constant.

i/p size $= n$. $O(n^2) \equiv$ poly-time.

for i/p size = n , $O(n^2) \equiv$ poly-

Subsetsum

Tuesday, March 10, 2026 9:20 AM

☆ Subset Sum

Given a set of positive integers.

$A = \{a_1, a_2, \ldots, a_n\}$

& target T (positive integer) → TRUE or FALSE

Check if $\exists S \subseteq A$ that sum to T.

$$\sum_{a \in S} a = T$$

e.g. $A = \{10, 1, 2, 5, 7, 13, 11\}$ $T = 22$

Ans: TRUE

- $\{10, 1, 11\}$
- $\{2, 7, 13\}$
- $\{10, 5, 7\}$

① Subproblem Def:

$a_1, \ldots, a_n$ as a seq.

① Subproblem Def:

Intuition: think $A = a_1, a_2, \ldots, a_n$ as a seq.

We want to pick a subseq that sums to T.

10, 1, 2, 5, 7, 13, [11] → take?

→ skip?

$SS(i, t)$ = TRUE if $\exists$ subset $\{a_1, \ldots, a_i\}$ that sums to t.

= FALSE o.w.

$\forall\ i = 0, 1, \ldots, n$, $t = 0, 1, \ldots, n$

where $SS(0, t)$ = TRUE if $\exists$ subset $\{\}$ that sums to t.

= FALSE o.w.

② Recursive Formula:

$SS(0,0)$ = TRUE

T

(2) Recursive

Base case $SS(0,0) = TRUE$

$SS(0,t) = FALSE \quad t = 1, \ldots, T$

$SS(i,0) = TRUE$ ✓

$SS(i,t) = SS((i-1),t) \vee SS((i-1), t-q_i)$

skip take

if $a_i \leq t$.

o.w.

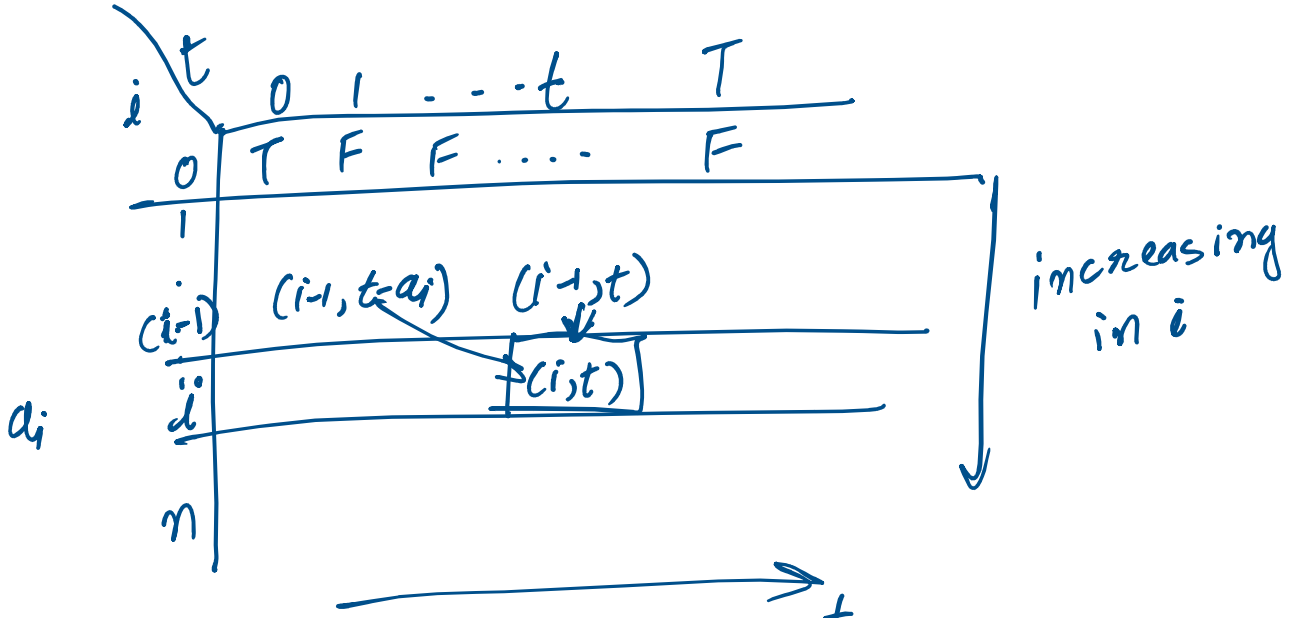
$= SS((i-1),t)$

Ans: $SS(n,T)$

☆ (Memorization) Evaluation Order.

$S[i,t] = SS(i,t) \quad i = 0,1,\ldots,n$

$t = 0,1,\ldots,T.$



Iterative Alg'

```
:
for i=1...n
  for t=0...T
:
```

Increasing in i; within that increasing in t.

Runtime: $(n*T)\ O(1) = O(n*T) \stackrel{?}{=} O(\text{poly}(n, \log_2 T))$ NO!

\# s.p.

i/p size : $|\{a_1, \ldots, a_n\}| + |T|$

$\ldots$ bits + $\log_2 T$ bits

i/p size : $\geq$ n bits + $\log_2 T$ bits

$= (n+a)$ bits

where $a = \log_2 T$

Now $a = \log_2 T \Rightarrow T = 2^{\log_2 T} = 2^a$.

Then, $O(n*T) = O(n * 2^{\log_2 T})$

$= O(n * 2^a)$

is exponential in a

$\Rightarrow$ exponential in i/p size

★ Question:

$\exists$ Alg of run time $O(n * \log_2 T)!$

poly-time $\equiv O((n * \log_2 T)^c)!$ for some c

or any poly-time $\equiv O(($ for constant c (e.g. $c = 10^6$!!!)

$\Big\|$

$P = NP$?