Today

- General announcement:

  Next Thursday department event:

  4-5pm: panel

  5-6pm: networking + food

  in new career services space.

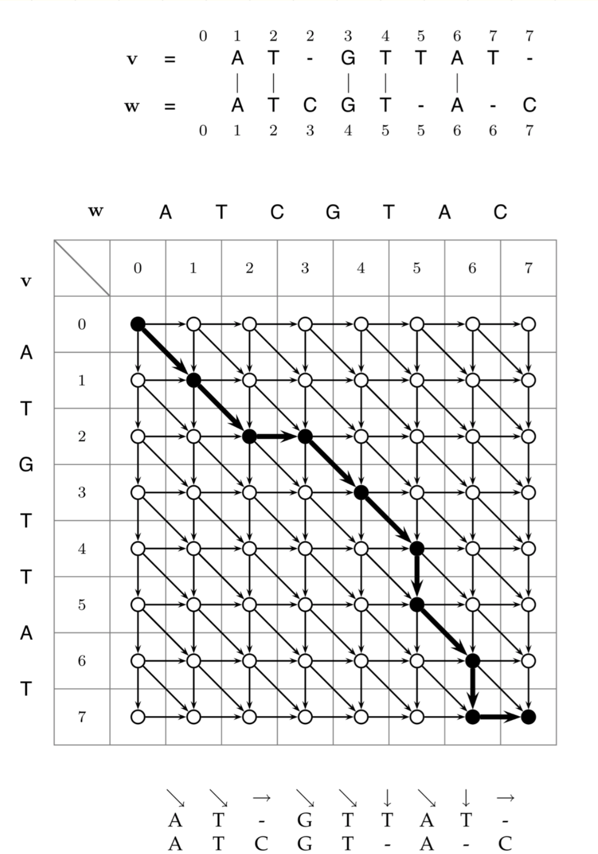
- HW2 graded + entered.

  Question: Who worked alone but is interested in a partner?

- HW3: due Monday

Last time: Edit distance

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



Next: Subset Sum

Given a set X of positive integers and a target value t, is there a subset of X which sums to t?

Recall our (exponential) backtracking.

Formalize this: recursion!

$$T(\underbrace{X}_{X[1..n]}, t) = \begin{cases} \text{include } X[1]: \\ T(X[2..n], t - X[1]) \\ \text{not}: \\ T(X[2..n], t) \end{cases}$$

```
SubsetSum(X[1..n], T):
    if T = 0
        return True
    else if T < 0 or n = 0
        return False
    else
        return (SubsetSum(X[1..n−1], T) ∨ SubsetSum(X[1..n−1], T − X[n]))
```

Led to exponential time:

Why?

Can we do DP?

In this chapter: reformulate
(& don't pass array, just index i)

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

Or:

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

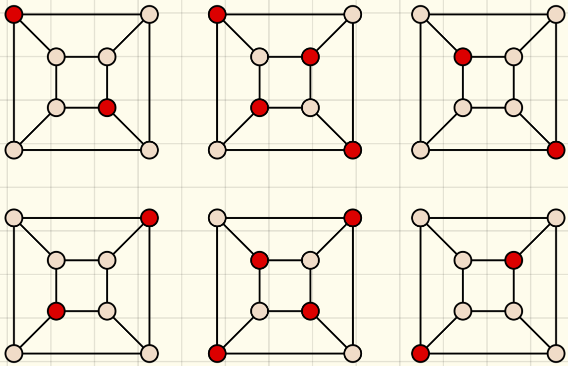
How to memoize?

Size / Space:

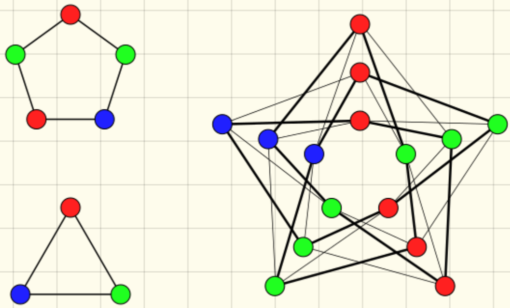
DP on trees:

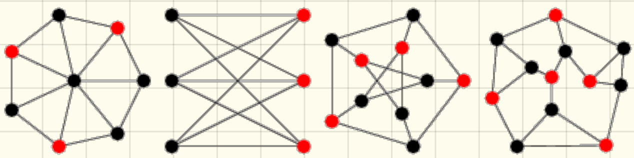
Consider independent set in a graph:

Can be many!



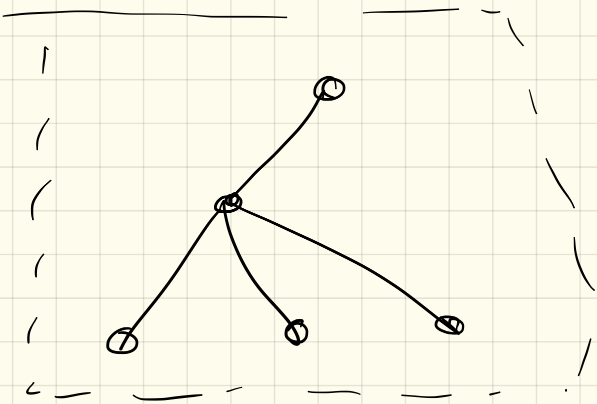
How to find?





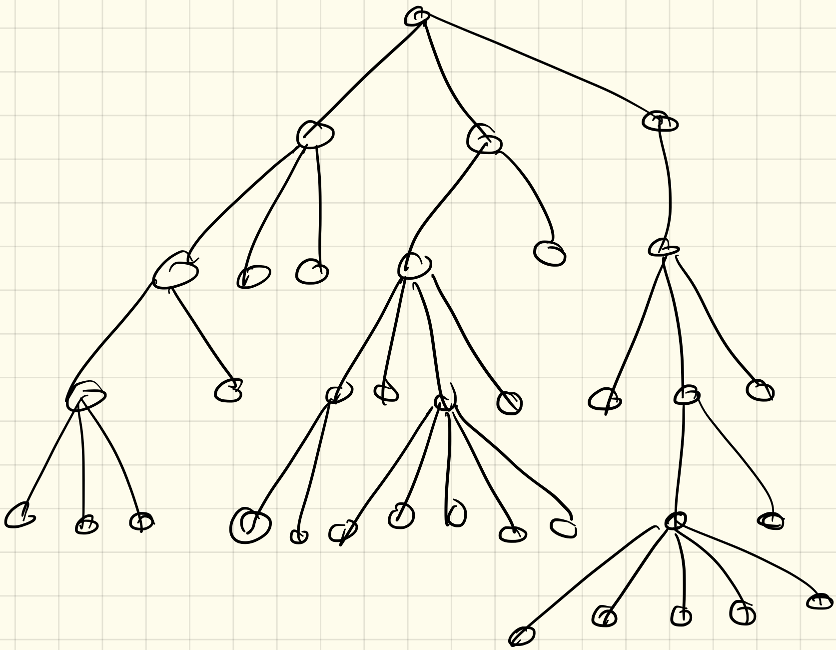
Well ... hard.

But! on trees, nice property:



Can make a semi-local choice.

Let's set up more carefully on a tree.



Still can't be greedy – but can try both "in" & "out" of IS for each vertex!

Let MISyes(v) = max size of ind. set in v's subtree is v is included

MISNo(v) = max size if v not included

What data structure will help us?

Then!

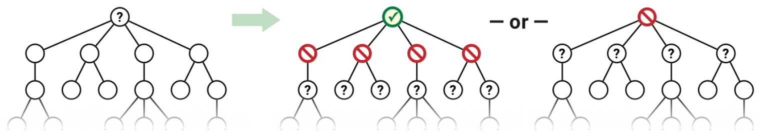


Figure 3.5. Computing the maximum independent set in a tree

Recursion

$$MISyes(v) = 1 + \sum_{w \downarrow v} MISno(w)$$

$$MISno(v) = \sum_{w \downarrow v} \max\{MISyes(w), MISno(w)\}$$

```
TREEMIS2(v):
  v.MISno ← 0
  v.MISyes ← 1
  for each child w of v
      v.MISno ← v.MISno + TREEMIS2(w)
      v.MISyes ← v.MISyes + w.MISno
  return max{v.MISyes, v.MISno}
```

Data Structure :