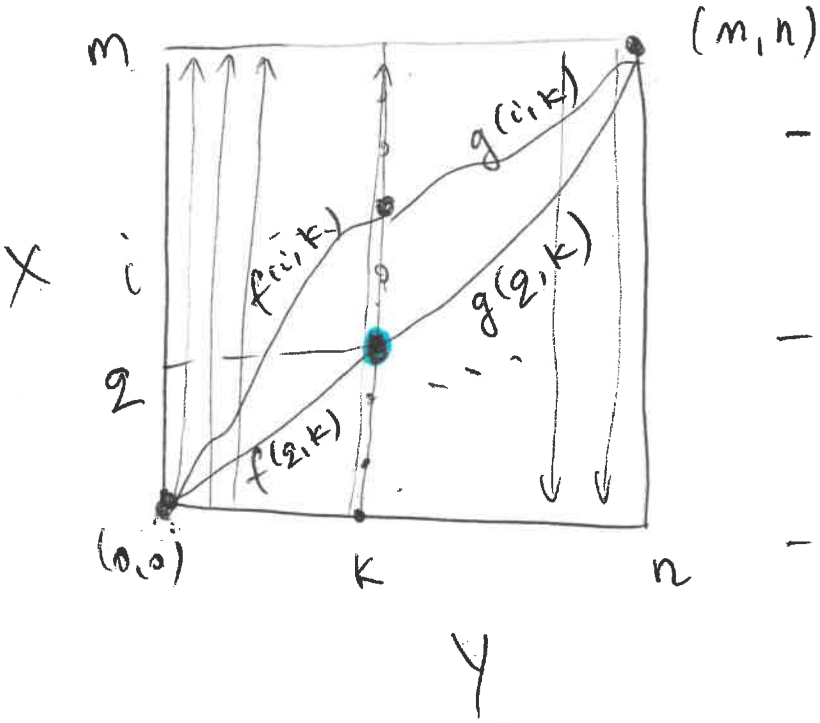
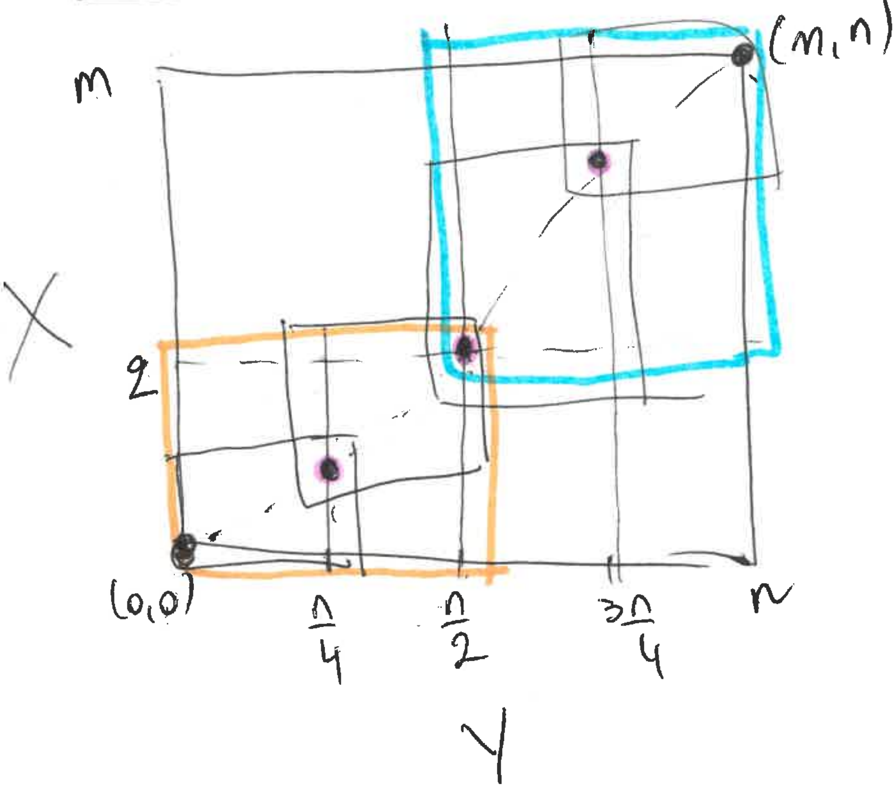
4.7.2017



- compute $f(i,k)$ for every index i
- compute $g(i,k)$ for every index i
- for every inde i compute $f(i,k) + g(i,k)$
- let q be the index for which $f(q,k) + g(q,k)$ is minimized

Then (q,k) is a point on the shortest-path from $(0,0)$ to (m,n) (is also part of an optimal alignment)

Divide-and-conquer



- assume that n is a power of 2

$T(m,n)$

- **Divide**

  $O(m \cdot n)$

- **Conquer** – 2 subproblems

  $T(2, \frac{n}{2}) + T(m-2, \frac{n}{2})$

- **Combine** – nothing

$$\boxed{T(m,n) = \underbrace{O(m \cdot n)}_{\text{divide}} + \underbrace{T(2, \tfrac{n}{2}) + T(m-2, \tfrac{n}{2})}_{\text{conquer}}}$$

$$\boxed{T(m,n) = O(m \cdot n)}$$

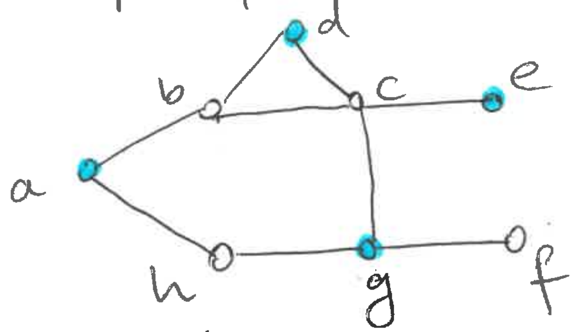
$$\boxed{\text{space} = O(m+n)}$$ linear!

Independent Set (IS) problem

An independent-set (IS) of a graph $G=(V,E)$ is a subset $V' \subseteq V$ s.t. each edge in E is incident to at most one vertex in V'.

The independent set problem asks to find a max-size independent set of a given (undirected) graph G.

- NP-complete problem



not an IS

independent sets : $\{a\}, \ldots, \cancel{\{a, h, g, f\}}, \{a, d, g\}, \ldots$

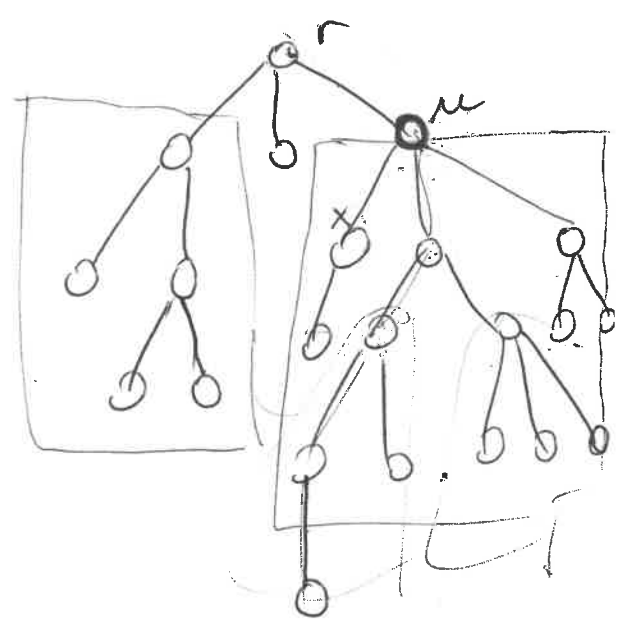
$\{a, d, g, e\}, \{b, h, e, f\}, \ldots$

Brute Force let $n = |V|$ $O(2^n \cdot n^2)$

- If the graph G is a tree, then we can solve the independent set problem using dynamic programming in polynomial time

- start by selecting the root r

$I(u)$ = size of a max-size independent-set of the subtree rooted at node u



Final objective: $I(r)$

DP proceeds from smaller subproblems to larger ones, in increasing order of the height of the node u.

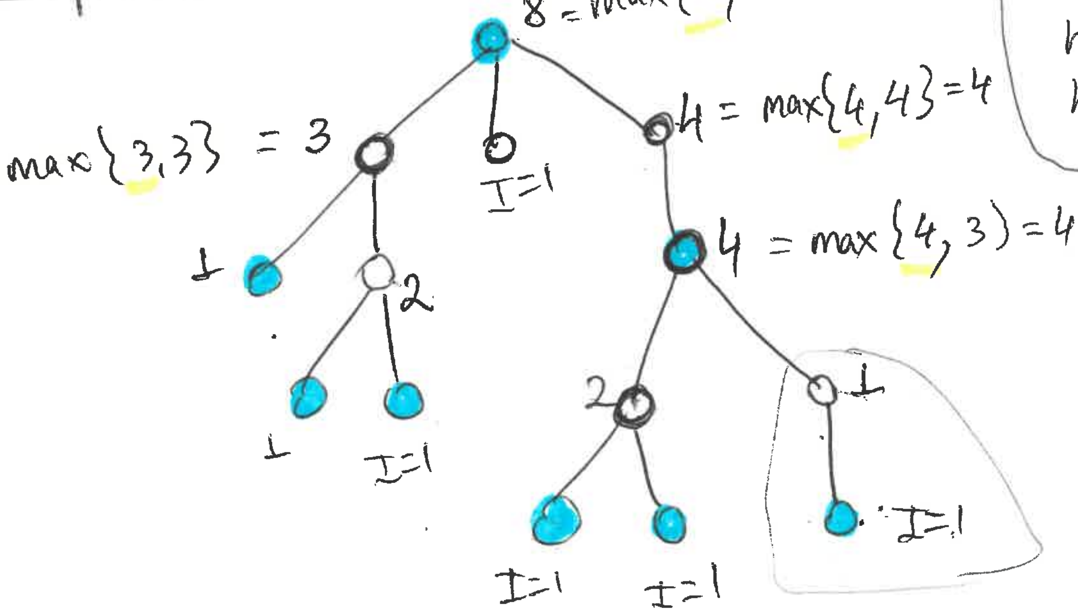
How do we compute $I(u)$?

$$\boxed{I(u) = \max\Big\{1 + \sum_{w \text{ is a grandchild of } u} I(w) \;,\; \sum_{v \text{ is a child of } u} I(v)\Big\}}$$

$\boxed{RT = O(n^2)}$

h = 0 ✓
h = 1 ✓
h = 2 ✓
h = 3 ✓
h = 4 ✓

<u>example</u>



max (1+0, 1+1) = 2

size of a maximum Independent Set is 8

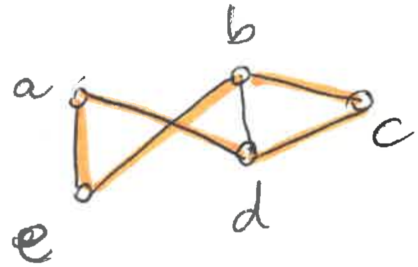
class NP

Hamiltonian-cycle problem (HC)

: Given an undirected graph $G=(V,E)$ find whether G has a hamiltonian cycle, that is a simple cycle that contains each vertex in G.

$n=|V|$

certificate : $\langle \overrightarrow{v_1, v_2, \ldots, v_n} \rangle$



$\langle a, b, c, d, e \rangle$ ✗

$\langle e, b, e, d, c \rangle$ ✗

$\langle c, d, a, e, b \rangle$ ✓

HC ∈ NP

- check if $(v_i, v_{i+1}) \in E \quad i=1,\ldots,n-1$
  $(v_n, v_1) \in E$
- check that each vertex occurs exactly once
- length of the list

Graph representation	
adj. matrix	adj. lists
$O(n)$	$O(n^2)$
$O(n)$	$O(n)$
$O(n)$	$O(n)$
RT $= O(n)$	$O(n^2)$

flag array