Michael A. Forbes
mforbes@illinois.edu
2024-01-29.1

CS473 Algorithms: Lecture 5 (2024-01-30)

logistics = - pset 1 due F17

last lecture = dynamic programming - abstract - memoization vs iterative
- trees vs DAG [naive] [strings subproblem soln]
- knapsack + param subproblem w/ $\boxed{\text{new}}$ variable
- pseudo poly vs poly time

diff lect4

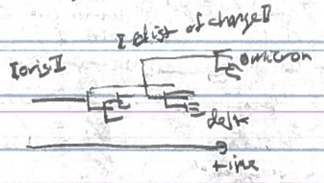
today - dynamic programming

reading = KT 6.6, 6.7

Q : how is covid evolving?

coronavirus genomes are RNA [ribonucleic acid] = strings over A, G, C, U

[evolution tree]
[important: rate of evolution = level of virality]



idea - place new strains as children of most similar known variant

Q : define similarity?

covid [actual coronavirus]

eg = cov1d
cove1d
covel
covel
3 changes [substitution]

vaccine [magazine articles on vaccines]

eg = vaccine
maccine
magzine
magazine [insertion!] [models of RNA mutation]
[deletion also]

def : $x, y \in \Sigma^*$ strings [str] over alphabet Σ. Their edit distance, $dist(x,y)$, is min # of insertions, deletions, substitutions required to transform x to y.

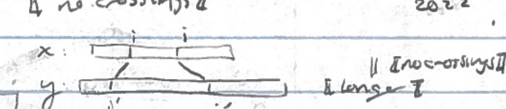
lem : $dist(x,y) = dist(y,x)$ [clear]

[alternate view] [convention]

def : $x, y \in \Sigma^*$, $n = |x|$, $m = |y|$. alignment of x, y is $A \subseteq [n] \times [m]$

st for $(i,j) \neq (i',j') \in A$, either $i < i'$, $j < j'$ or $i > i'$, $j > j'$ [no crossings]

x_i matched to y_j



cost of A is $|\{(i,j) \in A : x_i \neq y_j\}|$ [sub] $+ |\{i : x_i \text{ unmatched in } A\}|$ [del] $+ |\{j : y_j \text{ unmatched in } A\}|$ [insert]

eg = Vaccine
magazine
substitution
insertion

2022 mforbes.jm

fact : $dist(x,y)$ [def intuitive] $=$ min cost of x, y alignment [formal, less intuitive] [also friendly]

face: implications for left to right vs general editing

Q = compute edit distance efficiently? [brute force suffices] [drop brute force discussion] [dynamic programming]

Q : what are the subproblems? [align substrings]

vaccine
magazine

idea - compute edit dist between all contiguous substrings of x, and all substrings of y

Michael A. Forbes
miforbes@illinois.edu
2024-01-29.2

- 2022
- board instructions
- 2022 mforbes ism

lem. alignment of x,y A. if x_n, y_m both matched in [possibly separately] $\Rightarrow (x_n, y_m) \in A$ [matched to each other]

pf. x_n matched to $y_k \Rightarrow (n,k) \in A$ [non-crossing] $\Rightarrow$ $m \le k$, $\ell \le n$ [non-crossing] $\Rightarrow$ $k = m$, $k \le m$

[write] = y_m ——"—— $x_\ell \Rightarrow (\ell, m) \in A$

[decompose feasible solution space] [seen before]

cor. {alignments of x,y} =

{alignments of x,y, $(x_n, y_m) \in A$} = { $A \circ (x_n, y_m)$: A align $x_{<n}, y_{<m}$ } [prefix notation]

$\cup$ { ——"——, x_n unmatch} = { A : A align $x_{<n}, y$ } [x_n delete] [$\Rightarrow$: split off (x_n, y_m)] [$\Leftarrow$: merge (x_n, y_m)]

$\cup$ { ——"——, y_m ——"—— } = { A : A align $x, y_{<m}$ } [insert y_m]

[lem $\Rightarrow$ if x_n, y_m matched at all, matched to each other]

cor: [opt both sides]
$$\mathrm{dist}(x,y) = \min \begin{cases} \mathrm{dist}(x_{<n}, y_{<m}) + \mathbb{1}[x_n \ne y_m] & \text{[substitution] [indicator]} \\ \mathrm{dist}(x_{<n}, y) + 1 & \text{[deletion]} \\ \mathrm{dist}(x, y_{<m}) + 1 & \text{[insertion]} \end{cases}$$

exists optimal align of x,y w/ x_n, y_m matched iff / x_n deleted iff / y_n deleted iff [clear]

$\exists$ arg min / ——"—— / ——"——

- [new board]

lem: [base case] ε empty string. $\mathrm{dist}(x, \varepsilon) = |x|$, $\mathrm{dist}(\varepsilon, y) = |y|$

prop: $\mathrm{dist}(x,y)$ computable in $O(nm)$ time

- more idea before pf

pf. idea: subproblems $\mathrm{dist}(x_{\le i}, y_{\le j})$ [all prefixes] [small number] [support recursion]

algo:
- for $0 \le i \le n$: $d[i][0] = i$ [base cases] — $O(n)$
- for $0 \le j \le m$: $d[0][j] = j$ — $O(m)$
- for $1 \le i \le n$ — $O(n)$
  - for $1 \le j \le m$ — $O(m)$
    - $d[i][j] = \min \begin{cases} d[i-1][j-1] + \mathbb{1}[x_i \ne y_j] \\ d[i-1][j] + 1 \\ d[i][j-1] + 1 \end{cases}$ — $O(1)$

    [each already computed]
- return $d[n][m]$

- vertical line

[induction/recursion]

correctness: $\mathrm{dist}(x_{\le i}, y_{\le j}) = d[i][j]$ [recurrence]

complexity: $O(nm)$ time

Q: $\boxed{\text{space}}$ complexity? [many measures of complexity]
[in practice is more limiting]
SI: RAM usage on computer

prop: ——"—— and $O(nm)$ space

pf. algo: ——"——
correctness: ——"——
complexity: $d[\cdot][\cdot]$ has $O(nm)$ entries, each $O(1)$ size [convention: #s are small]

- convention

Q: is this good?

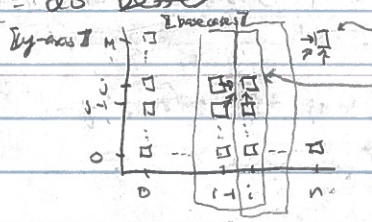
A: $\boxed{\text{yes}}$, polynomial

A: $\boxed{\text{no}}$, is impractical [n,m large in genomes] [$n \cdot m$ even larger]

- cite: space more expensive than time

Michael A. Forbes
miforbes@illinois.edu
2024-01-29.3

Q = do better?



dist(x,y)

$\mathrm{dist}(x_{\le i}, y_{\le j})$ [prefix] [subproblems] [relations]

board notes

[arrow structure]

idea - computing ith col, only need
- 0th row [base case]
- (i-1)st col [moving window]

[diagram] ⇒ $\boxed{\text{forget}}$ (i-2)st col

prop = dist(x,y) computable in $O(nm)$ time, $O(m)$ space

pf = **algo**:
- for $0 \le j \le m$ d[prev][j] = j ← 0th col [base case]
- for $1 \le i \le n$
  - d[cur][0] = i ← 0th row [base case]
  - for $1 \le j \le m$
    - d[cur][j] = min [as before] $\begin{cases} d[prev][j-1] + \mathbb{1}[x_i \ne y_j] & [cur-1] \\ d[prev][j] + 1 \\ d[cur][j-1] + 1 \end{cases}$ [all defined by now]
  - d[prev][·] ← d[cur][·] [move window]
  - unit d[cur][·] needed?
- return d[cur][m]

correctness = clear

complexity = time $O(nm)$ [as before]

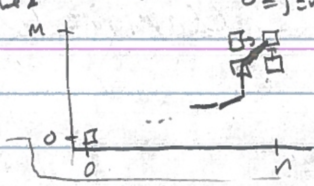
space d[cur][·], d[prev][·] $O(m)$ ☒

Q = computing the [alignment]?

2022 mforbes@jsm

prop - given $\{\mathrm{dist}(x_{\le i}, y_{\le j})\}_{0 \le i \le n,\ 0 \le j \le m}$ [as last lecture], can compute optimal alignment in $O(nm)$ space, time

sketch -



[chase argmin pointers, to base case]
[$O(n+m)$ steps]

do better.

☒

Q = compute alignment, in [small] space?

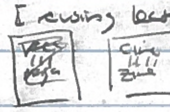
dist(x,y) in $O(m)$ space

[discrepancy]

align(x,y) via table of $O(nm)$ size

idea = divide and conquer
vaccine magazine
split [evenly] [general paradigm for optimizing complexity]
[how to use?]

idea = [reuse] space [reusing between time vs space]



[feasible soln decomp]

prop = $1 \le i \le n$, $\{\mathrm{align}(x,y)\} = \bigcup_{0 \le j \le m} \left\{ A_{\le} \circ A_{>} : \begin{array}{l} A_{\le} \text{ align } x_{\le i}, y_{\le j} \\ A_{>} \text{ align } x_{>i}, y_{>j} \end{array} \right\}$ [concat] [⇒: split] [⇐: merge]

[opt both sides]

cor $1 \le i \le n$ $\mathrm{dist}(x,y) = \min_j \{\mathrm{dist}(x_{\le i}, y_{\le j}) + \mathrm{dist}(x_{>i}, y_{>j})\}$

Michael A. Forbes
miforbes@illinois.edu
2024-01-29.4

[[restate]]

pg: $1 \le i \le n$ fixed. $\{dist(x_{\le i}, y_{\le j})\}_{0 \le j \le m}$ computable $O(nm)$ time $O(m)$ space [[space saving ok here]]

———— $\shortparallel$ ———— $x_{>i}, y_{\ge j}$ ———— $\shortparallel$ ———— [[symmetry]]

- diag

cor: $1 \le i \le n$ fixed. $meet_i(x,y) = \arg\min_j dist(x_{\le i}, y_{\le j}) + dist(x_{>i}, y_{>j})$

computable in $O(nm)$ time, $O(m)$ space

max $n+m$

pg: optimal alignment in $O(n \cdot m)$ time, $O(n+m)$ space

[[non space saving]]

pf: algo: align-concise(x,y) =
- if $n=1$, return align(x,y)
- if $m=1$, ———— $\shortparallel$ ————
- $j^* = meet_{n/2}(x,y)$
- $A_\le$ = align-concise$(x_{\le n/2}, y_{\le j^*})$
- $A_>$ = align-concise$(A_{>n/2}, y_{>j^*})$
- return $A_\le \circ A_>$

correctness: clear

complexity - space - [[reuse space]] [[meet, $A_\le, A_>$]]

$S(n,m) \le \max\{O(n+m), S(\frac{n}{2}, j^*), S(\frac{n}{2}, m-j^*)\}$

$\le O(n+m)$ via guess and check

time - $T(n,m) \le \underbrace{O(nm)}_{\alpha nm}^{[[meet]]} + T(n/2, j^*) + T(n/2, m-j^*)$

[[$\boxed{\text{two}}$ recursion]]

- rigor

$\boxed{\text{guess}}$ $T(n,m) \le \beta \cdot nm$

$\le \alpha \cdot nm + \underbrace{\beta \frac{n}{2} \cdot j^* + \beta \cdot \frac{n}{2}(m-j^*)}_{\le \beta/2\, nm}$

$\le \beta nm$ $\boxed{\text{if}}$ $\beta \ge 2\alpha$ [[works]] ☒

---

today - dynamic programming: edit distance - $O(nm)$ time $O(m)$ space - value / - alignment

- $O(n \cdot m)$ time $O(n+m)$ space - value [[$\boxed{\text{forward}}$ array]] / - alignment [[divide and conquer]]

reading - KT 6.6, 6.7

next lecture - DP

logistics - pset 1 due F1/7