

ALGORITHM DESIGN TECHNIQUES

Dynamic Programming : String / Text Problems: Examples

- Approximate String Matching: Edit Distance

PROBLEM – COMPUTING EDIT DISTANCE

- Consider the problem of finding occurrences of a given word w in a given text T :
 - this would require matching of strings
 - i.e. matching w with a text window $T[i..j]$
- Consider a word processor where the text is being edited and spelling errors are likely:
 - i.e. *matching has to be approximate*
- In such a context it is better for matching to compute a score – referred to as the *edit distance* – that indicates (dis)similarity:
 - for example the following strings would have an edit distance of 3:

○ G A ? C G G A T T A G C
 ↑ ↑ ↑
 ↓ ↓ ↓
○ G A T C G G A A T A G ?

PROBLEM – COMPUTING EDIT DISTANCE

- Problem Definition:

- Given strings S and T:
 - compute the number of insertions, deletions, or substitutions (of characters)
 - required on S so that it becomes T.

PROBLEM – COMPUTING EDIT DISTANCE

○ Recurrence on $S[1..i], T[1..j]$

- $\text{edit}(s[1..i], t[1..j])$

$$= \underline{\text{min}} \{$$

$$\text{edit}(s[1..i], t[1..j-1]) + 1,$$

$$\text{edit}(s[1..i-1], t[1..j-1]) + ((s[i] == t[j]) ? 0 : 1),$$

$$\text{edit}(s[1..i-1], t[1..j]) + 1,$$

$$\} \quad \text{if } i \geq 1 \text{ and } j \geq 1$$

$$= 0 \text{ otherwise}$$

Exercise:

Compare / Contrast this recurrence with the corresponding one for Sequence Alignment