

Epilogue: A Field Guide to Algorithm Design

.

Field Guide to Algorithm Design (I)

- ① Try to avoid solving the problem from scratch.
[special case of shortest paths, graph search, etc.]
- ② Try to simplify the problem with for-free primitives.
[sorting, connected components, etc.]
- ③ Is the "obvious" solution (like exhaustive search) good enough?
- ④ Brainstorm multiple greedy algorithms.
- ⑤ Natural way to split into subproblems \Rightarrow try divide + conquer.

Field Guide to Algorithm Design (II)

- ⑥ Try dynamic programming. How must the solution be built up from solutions to smaller subproblems?
- ⑦ Keep an eye out for repeated computations \Rightarrow speed up using suitable data structure. [Remember Principle of Parsimony]
- ⑧ Try to speed up/simplify using randomization.
- ⑨ Try reducing an NP-hard problem to yours.
Decide whether to compromise on correctness or speed.
- ⑩ Compromise on correctness \Rightarrow iterate again over the algorithm design paradigm, starting with greedy algorithms.

Field Guide to Algorithm Design (III)

- ⑪ Consider local search. [to solve from scratch, or post-process]
- ⑫ Compromise on speed \Rightarrow revisit dynamic programming.
- ⑬ Try latest + greatest MIP or SAT solvers.