Algorithm Reductions

Reductions

1. Sorting
2. Graph Problem
   1. Path Finding
   2. Largest Connected Component
   3. Connected
   4. Networking
   5. Cuts (Dijkstra’s Prim’s Kruskal)
3. Hashing
   1. Counting Sort
   2. Radix Sort
4. Dynamic Programming
   1. ???
5. Divide and Conquer
   1. ???
6. Window
   1. Substrings
   2. Palindromes
7. Longest Increasing Subsequence

Reduction Representations

1. Graphs
   1. Adjacency List
   2. Adjacency Matrix
   3. Edge Connections
2. Hashing
   1. HashMap
   2. TreeMap (sorts keys)
   3. TreeSet (sorts keys)
   4. Adjacency Lists
3. Dynamic Programming
   1. ???

Reduction Tips

1. Graph
   1. Is the input already in the form of an adjacency matrix or an adjacency list?
   2. How can I mark areas as fully explored/visited/at their best value?
   3. Can I apply BFS/DFS on the existing data structure?
2. Sorting
   1. Can I make this problem easier by sorting the input?
3. Hashing
   1. ???
4. Dynamic Programming
   1. Is the problem a binary decision? (keep / don’t keep / knapsack problem)
   2. Is it easier to start from the beginning or from the end?
   3. Can the problem be made easier by sorting?
5. Window
   1. Is this problem asking me to pattern match?
   2. Is this a string problem / substring problem?
   3. Will Dynamic Programming take too long / is it too confusing?

Algorithms to review

1. Graphs : Prim’s, Dijkstra’s, Kruskal’s, Topological Sort
2. DP: KnapSack