EE360C ALGORITHMS - FALL 2018 - December 6 - Evening

Final is cumulative!

Dynamic Programming

- Optimal Substructure
- Memoization
- Weighted Interval Scheduling
- Knapsack
- LCS
- Bellman-Ford
- Misc

Network Flow

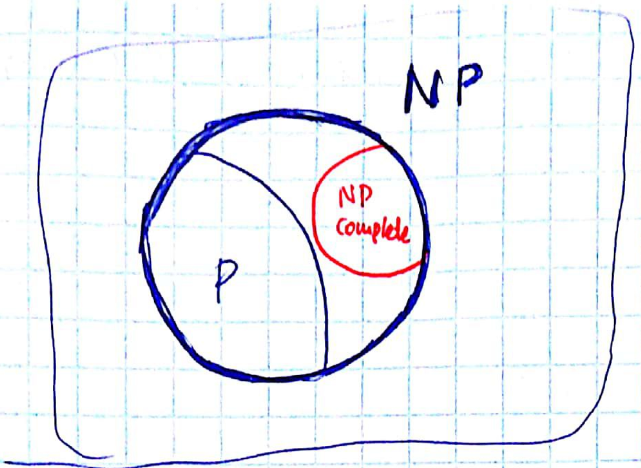
- Relationship between max flow and min-cut.
- Ford-Fulkerson algorithm
  - ↳ integer max flow (integer flow on every edge)
  - capacities = 1 ⇒ 0/1 max flow.
- Using network flow to solve other problems

NP-Completeness

- What is P, NP, NP-Complete?
- How to prove your problem is NP-Complete
  - ↳ poly-time reduction.

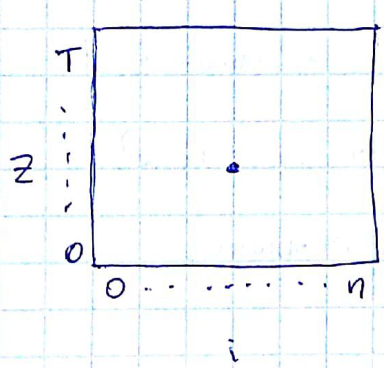
Subset Sum: Given positive integers $w_1, w_2, \ldots, w_n$ and a target T, is there a subset of the integers that sums to precisely T?

$$OPT(i, Z) = \begin{cases} 1, & \text{if } \exists \text{ subset of } w_i, \ldots, w_i \text{ that sums to } Z \\ 0, & \text{else} \end{cases}$$



(Subset-Sum, cont.)

$$OPT(i, z) = \begin{cases} \text{base cases} \\ \max\{ OPT(i-1, w), OPT(i-1, z-w_i) \} \end{cases}$$



$O(n \cdot T)$

if T is 2^n, it won't be poly-time

pseudo - polynomial time

pseudo: Polynomial in numeric value of the input

Subset-Sum is an NP-Complete problem.

$$C\text{-}SAT \leq_p \boxed{3\text{-}CNF\text{-}SAT \leq_p \text{SUBSET-SUM}}$$

CNF: conjuctive normal form

(_ OR _ OR _) AND (_ OR _ OR _) . . .

usual way in which it is shown that SUBSET-SUM is NP-Complete.