Lecture 1:

Tues Jan 29, 2019

What is an algorithm?

modified from CLRS

Def: An algorithm is a well-defined computational procedure that takes some value or set of values as input and produces some value or set of values as output.

An algorithm is a procedure for solving some well-defined computational problem.

Key Properties: Correct – that is, we can prove it solves the problem.

Efficient – We will spend effort formalizing this later, but we want the amount of computation (and space) to be polynomial in the length of the input.

Computational Problems (a 5 problem case study)

K+T (1.2)

- Seemingly similar problems, but with drastically different complexities.

(1) Interval Scheduling:

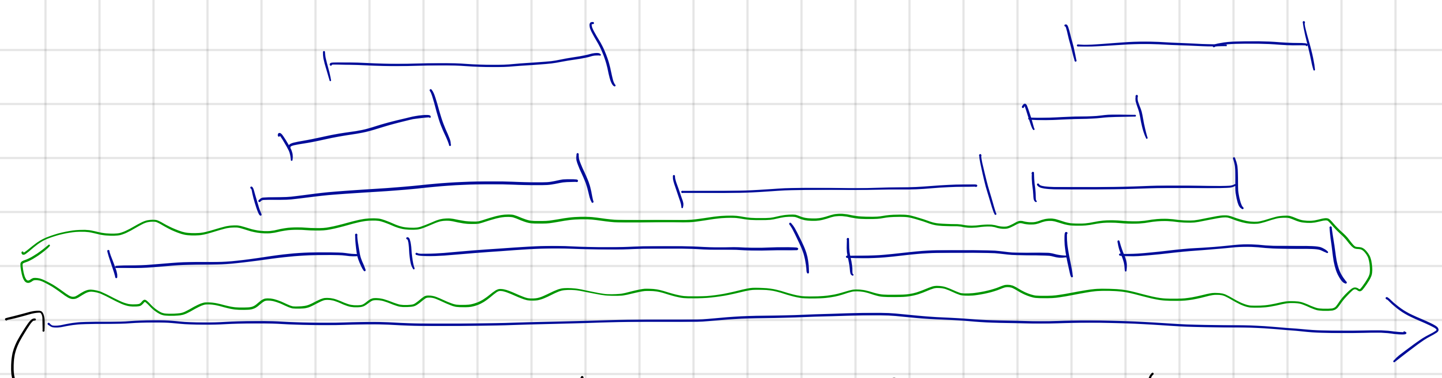
Given: $R = \{x_1, x_2, \ldots, x_n\}$, a set of requests such that only a single request can be accomodated at a particular time.

Find: The subset A ($A \subseteq R$) of requests that can be satisfied with the largest cardinality.

Def: Request x_i and x_j are <u>compatible</u> iff

$$start(x_j) \geq end(x_i) \text{ or } start(x_i) \geq end(x_j)$$

Def: A set A of requests is compatible if for all $x_i, x_j \in A$, with $x_i \neq x_j$, x_i and x_j are compatible.



Here, the maximum A is of cardinality 4 and there is only 1 subset of this size.

NOTE: In general, solutions are not unique.

- We will see that IS can be solved efficiently in $O(n \lg n)$ time, using a greedy approach.

(2) Weighted Interval Scheduling (WIS)

Same as IS except that, $\forall x_i \in R$, there is a weight $w_i > 0$.

Find: A subset $A \in R$ that is compatible and such that $\sum_{x_i \in A} w_i = \text{weight}(A)$ is maximum.

- the addition of weights changes the nature of the problem.

Assume $w_i > \sum_{j \neq i} w_j$. Then, any solution must include x_i regardless of the other things it contains.

- However, in the case that $w_1 = w_2 = \ldots = w_n$, we should obtain an algo. that solves IS.

- We say that WIS is a generalization of IS.

(3) Bipartite Matching

Def: <u>Bipartite</u>: A graph is bipartite if the vertex set can be decomposed as $V = X \cup Y$ such that $\forall e = \{u,v\} \in E$, either $u \in X$ and $v \in Y$ or $u \in Y$ and $v \in X$

Def: <u>Matching</u>: A matching in a graph $G=(V,E)$ is a subset of edges $M \subseteq E$ such that each node in V appears in <u>at most</u> one edge of M.

Given: A bipartite graph $G=(V,E)$.

Find: A matching $M \subseteq E$ of maximum size.

We will see how this can be solved via a process of <u>augmentation</u>; as an instance of a <u>network flow problem</u>. Complexity of $O(mn)$.

# of edges

# of nodes

(4) Independent Set (ISet)

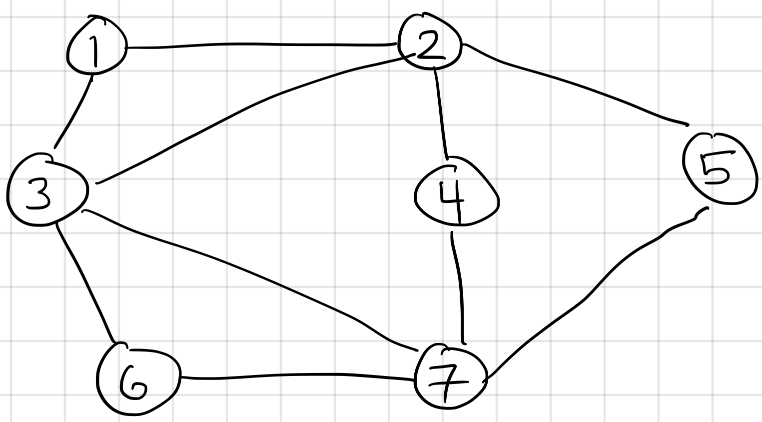
Given: A general, undirected graph $G = (V, E)$.

Find: A subset $S \subseteq V$ of V of maximum size such that S is independent.

Def: S is independent if $\forall\, u, v \in S$,

$\{u, v\} \notin E$.

Example:

$G =$ 

$\{1, 4, 5, 6\}$ is a MIS of G.

Q: MIS is strictly <u>more general</u> than IS or bipartite Matching, why?

A: We can represent IS & BM as instances of MIS.

How?

IS $\to$ MIS

Define $G = (V, E)$ where V is the set R of intervals and $E = \{ \{u, v\} \mid u, v \in R$ and u overlaps with $v \}$.

Independent Sets are just compatible subsets of intervals, and the MIS is the largest such set.

Q: What makes IS <u>easier</u> than MIS?

A: Special structure (order on the intervals).

BM $\to$ MIS

Given bipartite $G' = (V', E')$ define

$G = (V, E)$ where $V = E'$ (nodes of G are edges of G')

$E = \{ \{u, v\} \mid u, v \in V$ and u, v share an endpoint in $V' \}$

The MIS in G corresponds to a BM in G'

$\Rightarrow$ There are MIS instances that cannot arise as encodings of BM.

MIS is an **NP-complete** problem. We believe there is no algorithm to solve general (arbitrary) instances of MIS that run in time/space of the form $O(n^k)$ for some constant k.

But: Assume I want to prove to you that G contains some ISet of size m. Verifying this is easy. I give you the set S such that $|S| = m$, and you can verify if all vertices in S are independent.

there seems to be a fundamental difference between solving the problem & checking a solution to the problem.

(5) Competitive Facility Location

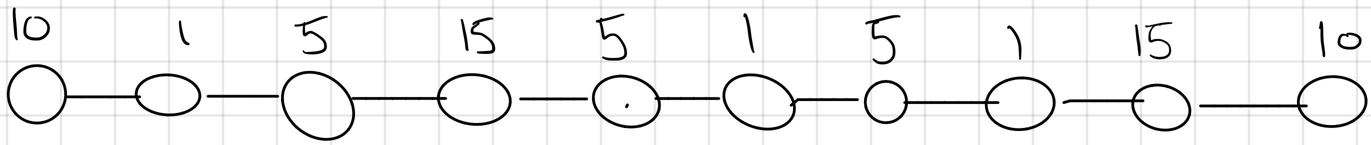
2 player game

Given a graph $G=(V,E)$ and a value function such that $\text{value}(v_i) = b_i$, Each player selects nodes to increase the overall value of the set of nodes they own. 2 adjacent nodes cannot be occupied (i.e. if $\{u,v\} \in E$ and u is occupied, a player cannot choose v).

Q: If 2 players alternate turns, and P_1 goes first, can P_2 obtain a set of vertices of some value B?

CFL is PSPACE-complete:

- Does not appear to be a trivial "verification" process for such problems.



Can we solve with $B = 20$? Yes

Can we solve with $B = 25$? NO

Not only is the <u>strategy</u> hard to derive, but even convincing /proving that a winning strategy exists is hard.

This is PSPACE-complete.

Believed to be strictly harder than NP-complete.

Many "games" reside in this category of problems.