Course : CSE 373 Analysis of Algorithms

Instructor : Rob Patro

Office : 259 **NEW** Computer Science

Office Hours : 10 - 11:30 AM Tu / Thu.

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:

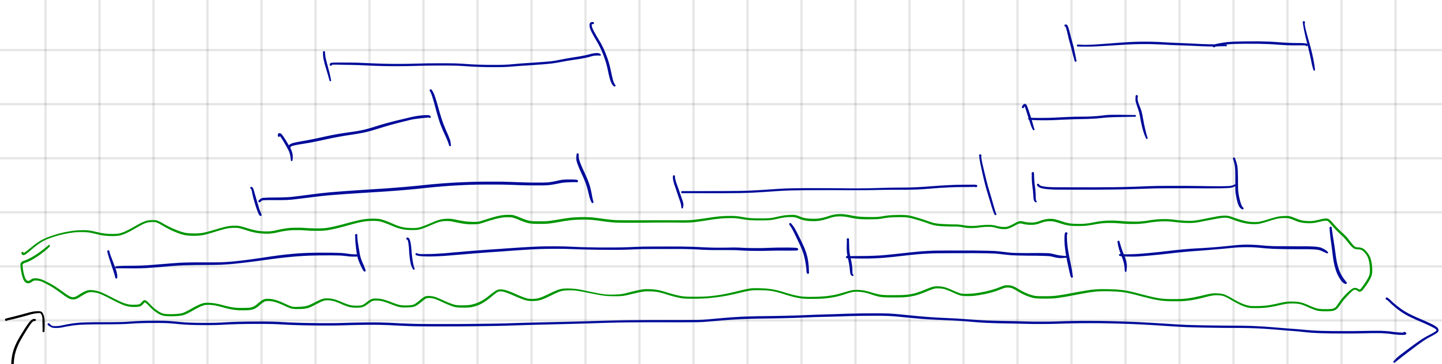
Given: $R = \{x_1, x_2, \ldots, x_n\}$, a set of requests such that only a single request can be accomodated at a particular time.

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

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

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 $w_1 = w_2 = \dots = w_n$, we should obtain an algo. that solves IS.

- We say that WIS is a generalization of IS.

(3) Bipartite Matching

Def: Bipartite : 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: Matching : A matching in a graph $G=(V,E)$ is a subset of edges $M \subseteq E$ such that each node in V appears in at most 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 augmentation, as an instance of a network flow problem. Complexity of $O(mn)$.

# of edges

# of nodes