Today

• Inductive Design of Algorithms

Inductive Design

• Objective: Define the solution for input of size $n+1$ in terms of solutions for inputs of size $\leq n$

Inductive Sort

<table>
<thead>
<tr>
<th>Transform Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Remove element $x$</td>
</tr>
<tr>
<td>• Remove max</td>
</tr>
<tr>
<td>• Remove min</td>
</tr>
<tr>
<td>• Partition input into 2 sets</td>
</tr>
<tr>
<td>• Partition input into 2 sets about pivot</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Transform output</th>
</tr>
</thead>
<tbody>
<tr>
<td>• ____________</td>
</tr>
<tr>
<td>• ____________</td>
</tr>
<tr>
<td>• ____________</td>
</tr>
<tr>
<td>• ____________</td>
</tr>
</tbody>
</table>
Induced Subgraphs

- Let \( G = (V, E) \) be a graph and let \( W \) be a subset of \( V \).
- The subgraph of \( G \) induced by \( W \) is the graph with
  - Vertex set: \( W \)
  - Edge set: \( \{ (x, y) \in E \mid x, y \in W \} \)

Maximal Induced Subgraph

- Input: A graph \( G = (V, E) \) and an integer \( k \).
- Output: A largest subgraph \( G' = (V', E') \) of \( G \) such that every vertex of \( G' \) has degree at least \( k \).

MIS\(_{n+1}\) Algorithm

Finds MIS on graphs with \( n+1 \) or fewer nodes.
MIS(G, k)

• If every vertex has degree at least k
  – Return G

• Else
  – Let x be a vertex with degree less than k
  – Let G' be the subgraph of G induced by V-{x}
  – Return MIS_n(G', k)

Correctness

Let H be a maximal induced subgraph of G with degree at least k.

– Case 1: Every vertex of G has degree at least k:
– Case 2: A vertex x of G has degree less than k:

Running Time

( Assume G has n vertices and m edges)

• Adjacency Matrix: \( T(n) \leq T(n-1) + cn^2 \)

• Adjacency List: \( T(n,m) \leq T(n-1,m-1) + m \)

• Other:

Polynomial Evaluation

• Input: Integers \( a_n, a_{n-1}, \ldots, a_0 \) and an integer x.
• Output: \( P_n(x) = \sum_{i=0}^{n} a_i x^i \)

Polynomial Evaluation Example

• Input: 3, -1, 0, 2 and 2
• Output: 22
• Explanation: \( P(x) = 3x^3 - x^2 + 2; \ P(2) = 22 \)

Naïve Approach

• Number of multiplications:
• Number of additions:
Pe$_{n+1}$ Algorithm
Evaluation of polynomials of degree at most $n+1$

PE(a$_n$,...,a$_0$,x)
- If $n=0$ then return $a_0$
- Else return $a_0 + x \cdot \text{PE}(a_n, ..., a_1, x)$

Inductive Approach
- Number of multiplications:
  $M(n) = 1 + M(n-1), M(0) = 0$
  Closed form: $M(n) = n$
- Number of additions:
  $A(n) = 1 + A(n-1), A(0) = 0$
  Closed form: $A(n) = n$

Vertex Cover
- Let $G=(V,E)$ be a graph
- A vertex cover of $G$ is a subset $W \subseteq V$ such that for every edge $e=(x,y)$ of $G$ either $x$ is in $W$ or $y$ is in $W$

Vertex Cover Example
Some vertex covers:
- $\{v_1, v_2, v_3, v_4\}$
- $\{v_2, v_3\}$

Vertex Covers in Forests
- Input: Forest $F=(V,E)$
- Output: A smallest vertex cover of $F$
A few observations and definitions

- A collection of trees is a forest
- A tree on \( n \) nodes has \( n - 1 \) edges
- A tree never has a cycle
- A tree is always connected
- A tree need not be rooted
- A node with 0 or 1 edges in a tree is a leaf
- A (non-empty) tree always has a leaf

Simple Case

- If \( F \) has no edges then return _________

Claim

- If \( u \) is a leaf of \( F \) and \( v \) is adjacent to \( u \) then some smallest vertex cover of \( F \) includes \( v \).
- Proof:
  - Let \( W \) be a smallest vertex cover of \( F \).
  - If \( v \) is not in \( W \) then \( u \) must be.
  - But then \( W \{-u\} + \{v\} \) is also a smallest vertex cover of \( F \).

Vertex Cover in Forests

If \( F \) has no edges return \( \emptyset \)
Else {
  Let \( u \) be a leaf with an edge in \( F \)
  Let \( v \) be the node adjacent to \( u \) in \( F \)
  Let \( F' \) be the subgraph of \( F \) induced by \( V \{-u,v\} \)
  Return \( \{v\} \cup VCT(F') \)
}