CS F364
Design & Analysis of Algorithms

ANALYSIS – PROBLEMS – REDUCTIONS

Analysis of Problems

- **Lower Bounds and Reduction: Example**

4/22/2016 Sundar B. CSIS, BITS, Pilani

Problem – Convex Hull

- The Convex Hull problem in two dimensions is defined as follows:
  - Given a set S of points <p_1, p_2, ... p_m> in 2-d space, find the smallest convex polygon that encloses all p_i in S.
    - A polygon encloses a point if the point is inside or on (one of) the edges of the polygon

Reduction - Example

- Example:
  - SORT $\precsim_{O(m)}$ CONVEX_HULL where m is the size of the input list.
- Understanding CONVEX_HULL:
  - CONVEX_HULL requires points in 2-D space as input.
  - The output of CONVEX_HULL is a list of vertices of a polygon

SORT $\precsim_{O(M)}$ CONVEX_HULL

- Intuition:
  - CONVEX_HULL requires points in 2-D space as input
    - i.e. we need a mapping from points in 1-D space to 2-D space:
      - we need a function f so that we can take the input to SORT – a list of values $<v_1,v_2,...,v_n>$ – and map it to a list of the form $<(v_1,f(v_1)), (v_2,f(v_2)), \ldots , (v_n,f(v_n))>$
  - The output of CONVEX_HULL is a list of vertices of a polygon

SORT $\precsim_{O(M)}$ CONVEX_HULL

- Intuition:
  - CONVEX_HULL requires points in 2-D space as input
  - The output of CONVEX_HULL is a list of vertices of a polygon
    - i.e. we need to ensure that
      - the output contains all the original points
      - and points can be extracted in order without too much additional cost

SORT $\precsim_{O(M)}$ CONVEX_HULL

- This forms a sorting algorithm:
  - Map the input list of points $L = <v_1,v_2,\ldots,v_n>$ to $L' = <(v_1, v_1^2), (v_2, v_2^2), \ldots, (v_n,v_n^2)>$
  - Apply a CONVEX_HULL algorithm on L' to get a permutation P of these points (as vertices of the polygon):
    - find p_i in P such that if $p_i = (v,v^2)$, then v is the minimum in L
  - Then output the points in the order p_i , $p_{(i+1)\%n}$, … $p_{(i+n-1)\%n}$
- Exercise:
  - Argue that the output is in sorted order.