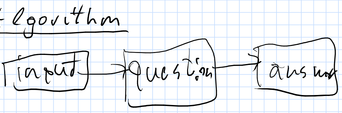
3/4 Lecture 10

Algorithm




- Decision problems. Y/N questions
- Search problems

  G: Graph, $s,t \in V(G)$

  - Is there a path between s and t in G.? [Reachability.]
  - Compute and output a path connecting s to t.
- Optimization problems

  Compute best solution in the solution space.

  Shortest path ≈ compute shortest path between s and t in G.

---

RAM Model (single core/thread etc).

---

Running time

n: input size

RT

$f(n)$ – upper bound on the # of operations performed by the alg for input of size n.

worst case analysis

$f(n) = O(n^2) \Rightarrow \exists c>0\ \forall n>0\ \ f(n) \le cn^2$

$f(n) = \Omega(n \log n)$ $\exists c>0\ \forall n\ \ f(n) \ge cn\log n$ for

$\Theta(n \log n)$

---

- Recursion
- Divide and conquer
- Graph algorithms.
- Greedy algorithms.

---

Reductions

Recursion

---

Uniqness

A[1....n] real numbers.

Q: Are all the numbers in A distinct?

Naive alg: Try all pairs $O(n^2)$.

Reduction to sorting

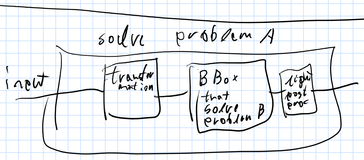
$O(n\log n)$ Sort the numbers.

$O(n)$ Check if two consecutive numbers are equal.

---

Thm If only comparisions are allowed UNIQNESS requires $\Omega(n \log n)$ time. Conditional lower bounds.

---




If we believe A cannot be solved efficiently

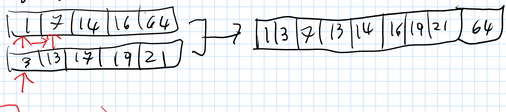
$\Rightarrow$ B can not be solved efficiently

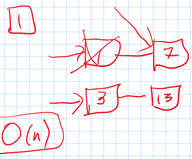
$A \le_p B$

---

Sorting and merging

Merging:

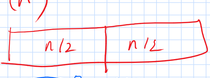



$O(n)$

---

SA $O(n^2)$




SB(A[1....n]):

SB(A[1....n/2])

SB(A[n/2+1....n])

merge(two lists)

Base cases

$SA(n) \ge n^2$ operations

Merge sort

$\left(\frac{n}{2}\right)^2 + \left(\frac{n}{2}\right)^2 + n = \frac{n^2}{4} + \frac{n^2}{4} + n$

$= \frac{n^2}{2} + n \le n^2$

$SA(n)$

$O(n \log n)$