

Note: this set of notes (and those for the next several lectures) are just my own notes that I used in class. We didn't necessarily cover everything on the notes on a particular day, so some material is repeated. Also, all the material is available in the Dasgupta book, chapters 3, 4, and 5, I believe. -jfh

somnolent
estuary
define
oersted
hijack
afghan
burst
canopy
laughing

Last time: DFS, BFS
<code handout>

For graphs, "linear time" means $O(|V| + |E|)$.
For simple graphs $|E|$ is $O(|V|^2)$

Warmup: give a linear-time algorithm for computing the reverse of a graph in adjacency-list format

Applications of DFS:

CONNECTED graphs; modify DFS to label each node with its conn comp number (pick numbers however you like).

Application: word ladders: COLD, CORD, CARD, CART, WART, WARM

CYCLE finding: note back-edges!

bipartite: colored by two colors.

Give linear time algorithm to tell whether undirected graph is bipartite

Fact: bipartite \Leftrightarrow no odd-length cycles

Give a linear time algorithm to find an odd length cycle in a strongly-connected directed graph (or report that none exists). (Strongly connected means that there's a directed path between any two verts)

Linear time algorithm: given an undirected graph G and an edge e , does G have a cycle containing e ?

Observation: in a dag, every edge leads to a vert with a lower "post" number!

Corr: every dag has a source and a sink (possibly many)

Topological sort: find a source; output it; delete from graph.
Ends up with order consistent with the DAG structure!

Appn: Brown CS courses.
