CS-170 Exam 2 March 22, 1995

David Wolfe

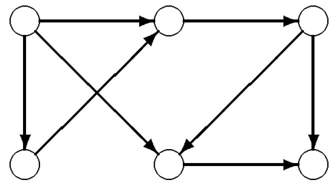
You should not need to write any code for this exam. Please make your answers as brief and as clear as possible. I highly recommend crossing out mistakes with a few dark strokes of the pen rather than erasing the work completely in case it is worth partial credit.

Below is a summary line for each question on the exam. You may use the backs of pages if you need more space, but please indicate for the grader you've done so: For example, "Continued on back of page 4". Please leave the exam stapled. You can look pick up a solution set (with the questions repeated) when you leave.

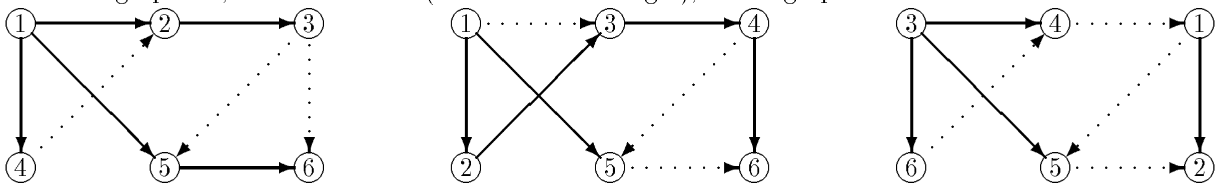
1.	short-answer graph problems	/42
2.	reachability	/20
3.	0-1-knapsack variant	/30
4.	kth largest in a heap	/30
Total	(Extra points possible)	/122

1. (7 points per part) **No words are required (nor recommended).** Consider the following graph G:

$G \Rightarrow$



Consider graphs A, B and C below (shown as **solid** edges), all subgraphs of G:



A B C

Some answers to parts a,b, and c below may be the same. Vertex numbers are for part (c) only.

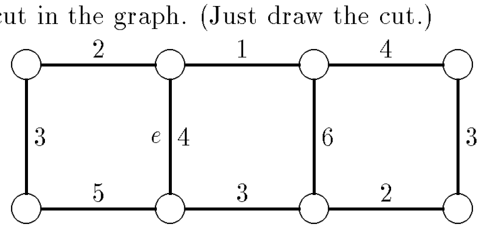
(a) Which **one** of the subgraphs indicated could be a breadth first search tree? ☐

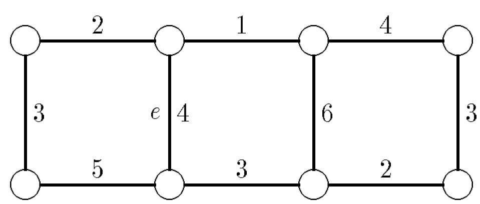
(b) Which **one** of the subgraphs could be a depth first search forest? ☐

(c) Which **one** of the graphs is labeled with a topological sort <u>of G</u>? ☐

(d) How many strongly connected components does original graph, G, have? ☐

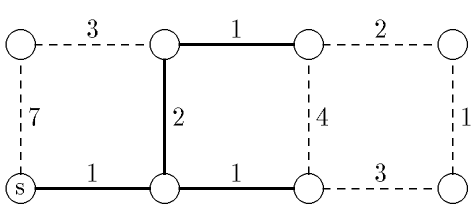
(e) For the following graph, prove edge e is part of **some** minimum spanning tree by drawing an appropriate cut in the graph. (Just draw the cut.)

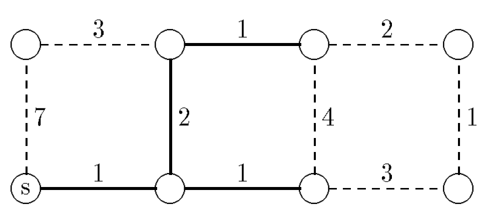




(Second copy is for your convenience.)

(f) Dijkstra's algorithm is being run to find shortest paths from source s in the following undirected graph. The graph is shown in dashed edges, and the tree so far is solid. Darken the dashed edge which would next be added to the tree.





(Second copy is for your convenience.)

2. (20 points) The all-pairs *reachability* problem: Given a directed graph $G = (V, E)$ in the form of an **adjacency matrix**, determine for each pair of vertices $u, v \in V$ if there is a path from u to v. Propose an efficient algorithm to solve this problem and analyze your algorithm.

   - full-credit for $O(V(E + V))$
   - extra-credit for anything faster, say $O(V^2)$ or $O(V^2 \log V)$

3. (30 points) Consider the following variant of the 0-1-knapsack problem in which there is an unlimited supply of each item. You have won a shopping spree at a store where there are n products; the i^{th} product has size w_i and value v_i. You have a grocery cart which can be filled with products whose size totals to W, taking as many of each product as you wish. Your goal is to maximize the total value of the products you take.

   (In the 0-1 knapsack problem, you take at most one of each product. In this problem, there is an unbounded supply of each product.)

   Determine two dynamic programming algorithms to the value of goods, V, which you can take during your shopping spree. It suffices to give a recurrence with a one line explanation verifying the running time of a dynamic programming algorithm to solve the recurrence.

   (a) Give an $O(nW)$ solution. (Hint: Let V_w be maximum value you can pack into a grocery cart holding total size w.)

   (b) Give an $O(nV)$ solution, where V is the total value you can pack in the grocery cart in the solution. (Hint: Let W_v be the minimum sized basket you need to hold a total value of v.)

4. (30 points) Give the best algorithm you can to find the k^{th}-smallest element in an n-node binary heap, where n is **much** larger than k. You'll receive:

- 1/2-credit for $O(k \log n)$
- 1/2-credit for $O(n)$; 5 extra points if it is in addition to another solution.
- 3/4-credit for an $O(2^k)$ algorithm.
- full-credit for $O(k^2)$
- extra-credit for $O(k \log k)$

MORE SPACE IF REQUIRED