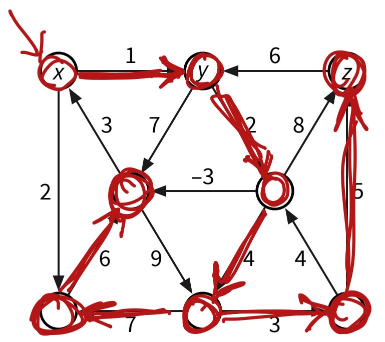
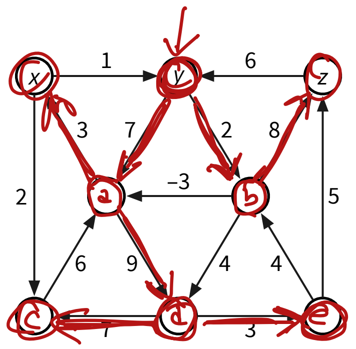
CS/ECE 374 A ✧ Fall 2021 Fake Midterm 2 Problem 1	Name:

Clearly indicate the following structures in the directed graph below, or write NONE if the indicated structure does not exist. Don't be subtle; to indicate a collection of edges, draw a heavy black line along the entire length of each edge.

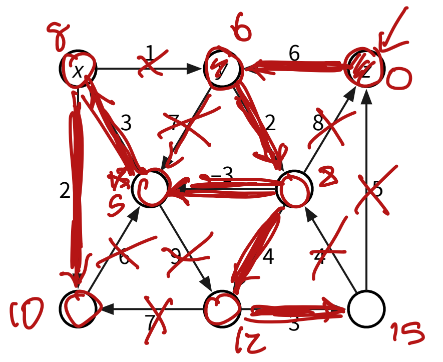


(a) A depth-first tree rooted at x.



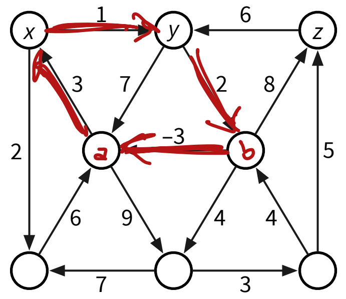
(b) A breadth-first tree rooted at y.

Queue: ~~a~~ ~~b~~ ~~d~~ x z e c

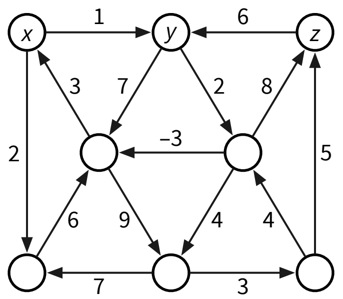


(c) A shortest-path tree rooted at z.

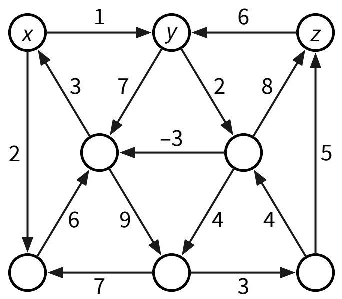
$\boxed{3}$



(d) The shortest directed cycle.



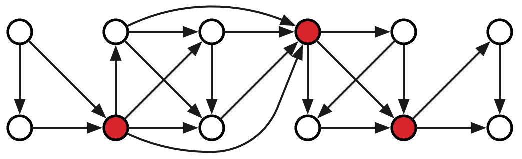
[scratch]

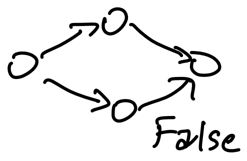


[scratch]

CS/ECE 374 A ✧ Fall 2021 Fake Midterm 2 Problem 2	Name:

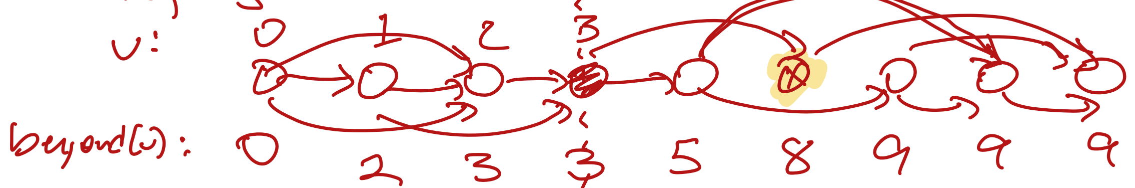
A vertex v in a (weakly) connected graph G is called a ***cut vertex*** if the subgraph $G - v$ is disconnected. For example, the following graph has three cut vertices, which are shaded in the figure.





Suppose you are given a (weakly) connected *dag* G with one source and one sink. Describe and analyze an algorithm that returns TRUE if G has a cut vertex and FALSE otherwise.

Topological sort — index vertices 1..V



beyond(v) = max{w | u→w is edge and u<v in top order}

$$\text{beyond}(v) = \begin{cases} 0 & \text{if } v \text{ source} \\ \max\left\{ \begin{array}{l} \max\{w \mid (v-1)\to w \text{ is edge}\} \\ \text{beyond}(v-1) \end{array} \right\} \end{cases}$$

memoize into G

eval in top order (increasing index $\longrightarrow$)

Top sort: $O(\cancel{V+}E)$ $\boxed{\text{Eval: } O(\cancel{V+}E) \text{ time}}$

$$\boxed{\boxed{O(V+E)}}$$

Find vertex v s.t. $v = \text{beyond}(v)$ but $v \neq$ ~~s~~source or sink

CS/ECE 374 A ✧ Fall 2021 Fake Midterm 2 Problem 3	Name:

You decide to take your next hiking trip in Jellystone National Park. You have a map of the park's trails that lists all the scenic views in the park, but also warns that certain trail segments have a high risk of bear encounters. To make the hike worthwhile, you want to see at least three scenic views. You also don't want to get eaten by a bear, so you are willing to hike along at most one high-bear-risk segment. Because the trails are narrow, each trail segment allows traffic in only one direction.

Your friend has converted the map into a directed graph $G = (V, E)$, where V is the set of intersections and E is the set of trail segments. A subset S of the edges are marked as *Scenic*; another subset B of the edges are marked as *high-Bear-risk*. You may assume that $S \cap B = \varnothing$. Each segment $e \in E$ is also labeled with a positive length $\ell(e)$ in miles. Your campsite appears on the map as a particular vertex $s \in V$, and the visitor center is another vertex $t \in V$.

Describe and analyze an algorithm to compute the $\boxed{\text{shortest hike}}$ from your campsite s to the visitor center t that includes *at least* three scenic trail segments and *at most* one high-bear-risk trail segment. You may assume such a hike exists.

Build new graph $G' = (V', E')$

$$V' = \underset{\substack{\uparrow \\ \text{location}}}{V} \times \underset{\substack{\uparrow \\ \#\text{scenic} \\ \text{hiked}}}{\{0, 1, 2, \geq 3\}} \times \underset{\substack{\uparrow \\ \#\text{bears}}}{\{0, 1\}}$$

$$\begin{aligned} E' = (v, \$, 0) &\longrightarrow (w, \$, 1) && \text{if } v \to w \in B \\ (v, \$, b) &\longrightarrow (w, \$+1, b) && \text{if } v \to w \in S \\ (v, \$, b) &\longrightarrow (w, \$, b) && \text{otherwise} \end{aligned}$$

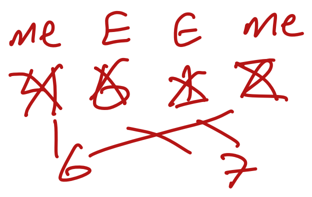
shortest from $(s, 0, 0)$ to $(t, \geq 3, \overset{0}{\underset{1}{\text{or}}})$

CS/ECE 374 A ✧ Fall 2021 **Fake Midterm 2 Problem 4**	Name:

During a family reunion over Thanksgiving break, your ultra-competitive thirteen-year-old nephew Elmo challenges you to a card game. At the beginning of the game, Elmo deals a long row of cards. Each card shows a number of points, which could be positive, negative, or zero. After the cards are dealt, you and Elmo alternate taking either the leftmost card or the rightmost card from the row, until all the cards are gone. The player that collects the most points is the winner.

For example, is the initial card values are $[4, 6, 1, 2]$, the game might proceed as follows:

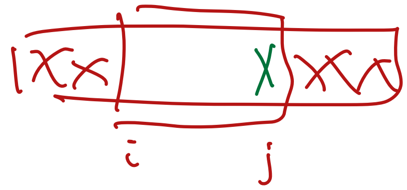
- You take the 4 on the left, leaving $[6, 1, 2]$.
- Elmo takes the 6 on the left, leaving $[1, 2]$.
- You take the 2 on the right, leaving $[1]$.
- Elmo takes the last 1, ending the game.
- You took $4 + 2 = 6$ points, and Elmo took $6 + 1 = 7$ points, so Elmo wins!



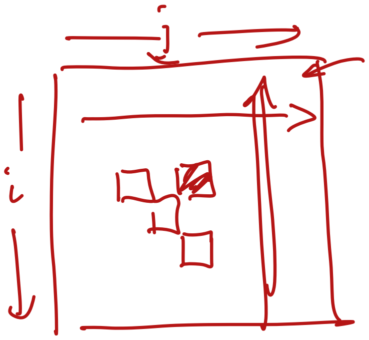
DP?

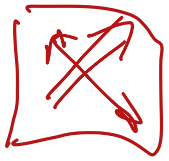
Describe and analyze an algorithm to determine, given the initial sequence of cards, the maximum number of points that you can collect playing against a *perfect* opponent. Assume that Elmo generously lets you move first.

MaxPoints(i, j) = max # points I can win starting with A[i..j] if it's my turn



$$MP(i,j) = \begin{cases} 0 & \text{if } j<i \\ A[i] & \text{if } j=i \\ \max\{A[i], A[i+1]\} & \text{if } j=i+1 \\ \max\begin{cases} A[i] + \min\{MP(i+2,j), MP(i+1,j-1)\} \\ A[j] + \min\{MP(i,j-2), MP(i+1,j-1)\} \end{cases} & \text{o/w} \end{cases}$$



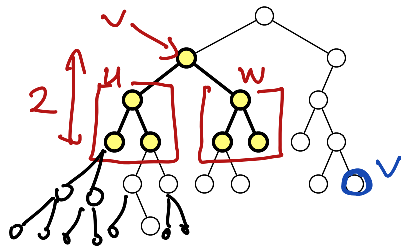


$O(n^2)$ time

CS/ECE 374 A ✧ Fall 2021 **Fake Midterm 2 Problem 5**	Name:

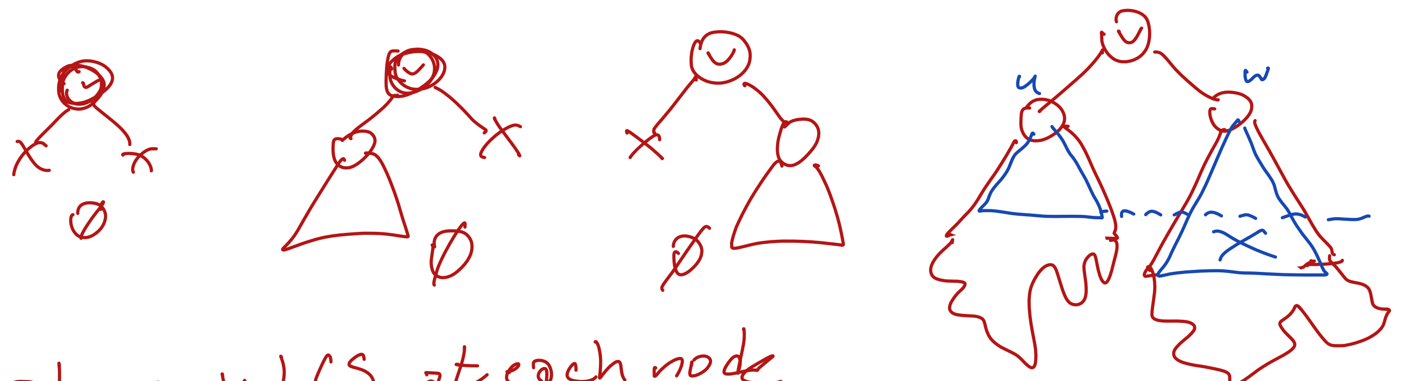
For this problem, a *subtree* of a binary tree means any connected subgraph. A binary tree is *complete* if every internal node has two children, and every leaf has exactly the same depth.

Describe and analyze a $\boxed{\text{recursive}}$ algorithm to compute the ***largest complete subtree*** of a given binary tree. Your algorithm should return both the root and the depth of this subtree. For example, given the following tree T as input, your algorithm should return the left child of the root of T and the integer 2.

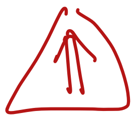


LCS(v) = depth of largest complete subtree rooted at v

$$LCS(v) = \begin{cases} 0 & \text{if } v \text{ is a leaf or has only one child} \\ \underline{1} + \min(LCS(v.left), LCS(v.right)) & \text{o/w} \end{cases}$$



store v.LCS at each node

eval  rev. level order

postorder

O(n) time to compute v.LCS ~~for~~ all v

return max