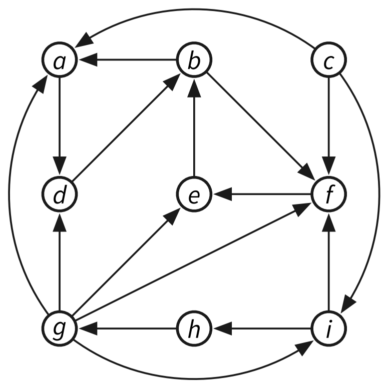
Write your answers in the separate answer booklet.

Please return this question sheet and your cheat sheet with your answers.

1. ***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 search tree rooted at vertex c.

(b) A breadth-first tree rooted at vertex c.

(c) The strong components of G. (Circle each strong component.)

(d) Draw the strong-component graph of G.

2. During her walk to work every morning, Rachel likes to buy a cappuccino at a local coffee shop, and a croissant at a local bakery. Her home town has *lots* of coffee shops and lots of bakeries, but strangely never in the same building. Punctuality is not Rachel's strongest trait, so to avoid losing her job, she wants to follow the shortest possible route.

Rachel has a map of her home town in the form of an undirected graph G, whose vertices represent intersections and whose edges represent roads between them. A subset of the vertices are marked as bakeries; another disjoint subset of vertices are marked as coffee shops. The graph has two special nodes s and t, which represent Rachel's home and work, respectively.

Describe an algorithm that computes the shortest path in G from s to t that visits both a bakery and a coffee shop, or correctly reports that no such path exists.

3. An undirected graph $G = (V, E)$ is ***bipartite*** if its vertices can be partitioned into two subsets L and R, such that every edge in E has one endpoint in L and one endpoint in R. Describe and analyze an algorithm to determine, given an undirected graph G as input, whether G is bipartite. *[Hint: Every tree is bipartite.]*

4. Satya is in charge of establishing a new testing center for the Standardized Awesomeness Test (SAT), and found an old conference hall that is perfect. The conference hall has n rooms of various sizes along a single long hallway, numbered in order from 1 through n. Each pair of adjacent rooms i and $i+1$ is separated by a single wall. Satya knows exactly how many students fit into each room, and he wants to use a subset of the rooms to host as many students as possible for testing.

   Unfortunately, there have been several incidents of students cheating at other testing centers by tapping secret codes through walls. To prevent this type of cheating, Satya can use two adjacent rooms only if he demolishes the wall between them. For example, if Satya wants to use rooms 1, 3, 4, 5, 7, 8, and 10, he must demolish three walls: between rooms 3 and 4, between rooms 4 and 5, and between rooms 7 and 8.

   The city's chief architect has determined that demolishing more than k walls would threaten the structural integrity of the building.

   Describe an efficient algorithm that computes the largest number of students that Satya can host for testing without demolishing more than k walls. The input to your algorithm is the integer k and an array $S[1\,..\,n]$, where each $S[i]$ is the (non-negative integer) number of students that can fit in room i.

5. Suppose you are given an array $A[1\,..\,n]$ of numbers.

   (a) Describe and analyze an algorithm that either returns two indices i and j such that $A[i]+A[j]=374$, or correctly reports that no such indices exist.

   (b) Describe and analyze an algorithm that either returns three indices i, j, and k such that $A[i]+A[j]+A[k]=374$, or correctly reports that no such indices exist.

   Do ***not*** use hashing. As always, faster correct algorithms are worth more points.