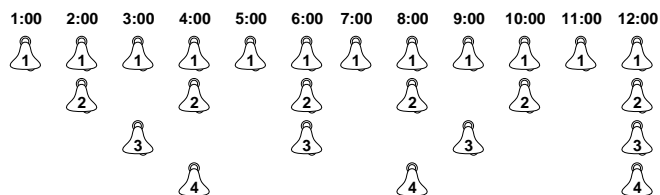


Write your answers in the separate answer booklet.
This is a 90-minute exam. The clock started when you got the questions.

1. Professor Quasimodo has built a device that automatically rings the bells in the tower of the Cathédrale de Notre Dame de Paris so he can finally visit his true love Esmerelda. Every hour exactly on the hour (when the minute hand is pointing at the 12), the device rings at least one of the n bells in the tower. Specifically, the i th bell is rung once every i hours.

For example, suppose $n = 4$. If Quasimodo starts his device just after midnight, then his device rings the bells according to the following twelve-hour schedule:



What is the *amortized* number of bells rung per hour, as a function of n ? For full credit, give an exact closed-form solution; a correct $\Theta()$ bound is worth 5 points.

2. Let G be a directed graph, where every edge $u \rightarrow v$ has a weight $w(u \rightarrow v)$. To compute the shortest paths from a start vertex s to every other node in the graph, the generic single-source shortest path algorithm calls INITSSSP once and then repeatedly calls RELAX until there are no more tense edges.

INITSSSP(s):
 $\text{dist}(s) \leftarrow 0$
 $\text{pred}(s) \leftarrow \text{NULL}$
 for all vertices $v \neq s$
 $\text{dist}(v) \leftarrow \infty$
 $\text{pred}(v) \leftarrow \text{NULL}$

RELAX($u \rightarrow v$):
 if $\text{dist}(v) > \text{dist}(u) + w(u \rightarrow v)$
 $\text{dist}(v) \leftarrow \text{dist}(u) + w(u \rightarrow v)$
 $\text{pred}(v) \leftarrow u$

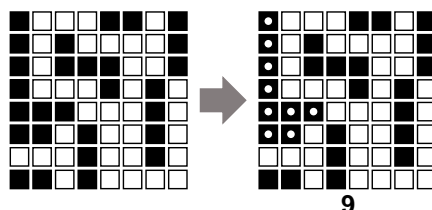
Suppose the input graph has no negative cycles. Let v be an arbitrary vertex in the input graph. **Prove** that after every call to RELAX, if $\text{dist}(v) \neq \infty$, then $\text{dist}(v)$ is the total weight of some path from s to v .

3. Suppose we want to maintain a dynamic set of values, subject to the following operations:
- INSERT(x): Add x to the set (if it isn't already there).
 - PRINT&DELETERANGE(a, b): Print and delete every element x in the range $a \leq x \leq b$. For example, if the current set is $\{1, 5, 3, 4, 8\}$, then PRINT&DELETERANGE(4, 6) prints the numbers 4 and 5 and changes the set to $\{1, 3, 8\}$.

Describe and analyze a data structure that supports these operations, each with amortized cost $O(\log n)$.

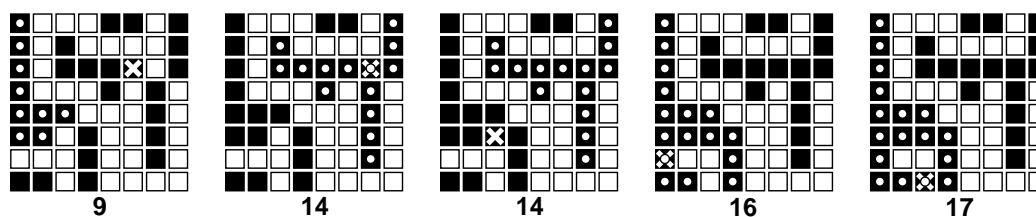
4. (a) [4 pts] Describe and analyze an algorithm to compute the size of the largest connected component of black pixels in an $n \times n$ bitmap $B[1..n, 1..n]$.

For example, given the bitmap below as input, your algorithm should return the number 9, because the largest connected black component (marked with white dots on the right) contains nine pixels.



- (b) [4 pts] Design and analyze an algorithm $\text{BLACKEN}(i, j)$ that colors the pixel $B[i, j]$ black and returns the size of the largest black component in the bitmap. For full credit, the *amortized* running time of your algorithm (starting with an all-white bitmap) must be as small as possible.

For example, at each step in the sequence below, we blacken the pixel marked with an X. The largest black component is marked with white dots; the number underneath shows the correct output of the BLACKEN algorithm.



- (c) [2 pts] What is the *worst-case* running time of your BLACKEN algorithm?

5. [Graduate students must answer this question.]

After a grueling 373 midterm, you decide to take the bus home. Since you planned ahead, you have a schedule that lists the times and locations of every stop of every bus in Champaign-Urbana. Unfortunately, there isn't a single bus that visits both your exam building and your home; you must transfer between bus lines at least once.

Describe and analyze an algorithm to determine the sequence of bus rides that will get you home as early as possible, assuming there are b different bus lines, and each bus stops n times per day. Your goal is to minimize your *arrival time*, not the time you actually spend travelling. Assume that the buses run exactly on schedule, that you have an accurate watch, and that you are too tired to walk between bus stops.