Solution

a) Implementation:

- $Create(n)$ = This $BitBlocks$ data structure will mirror the Union-Find implementation that uses pointers. So $Create(n)$ initializes a record for each bit i numbered 1-n where the name for each record of bit i is named i, and the value of each record is 0. Each record has a pointer that points to itself, otherwise defined as a $null$ pointer. We will also maintain an additional field that holds the size of each set of records. At first, this size will be 1 for every record.
- $SetToOne(i)$ = This method first finds the record with name i, and then updates its value to 1. It then checks the values of records $i + 1$. If this record has a value of 1, and i and $i + 1$ are not connected, the pointer of $i + 1$ is updated to point to i. We then check record $i - 1$. If it has a value of 1, and is not already connected to i, we update i's pointer to point to $i - 1$. With this implementation, the head of each set is the smallest i in the connected component. We update the size field associated with this record to be +1.
- $GetValue(i)$ = lookup the record named i and return its value.
- $GetBlockSize(i)$ = lookup the record named i and follow its pointers until you reach the head of its connected set (until the pointer points back to itself) and then retrieve the size value.

b) Proof: The implementations of each method are fairly straightforward. $Create(n)$ simply initializes n records with null pointers, so we know this works. $SetToOne(i)$ works by looking up record i, which we know we can do, and then updating the pointers of records i and/or $i+1$ as necessary. $GetValue(i)$ performs a simple lookup and retrieves the value of that record. $GetBlockSize(i)$ also performs a simple lookup, and then follows a trail of pointers to the head of the set, where it then retrieves a size value. We know this works because we structured our dataset such that in each connected component, all pointers go in the same direction, leading towards the lowest i.

c)
- $Create(n)$ makes n records, and so occupies space O(n) and time O(n).
- $SetToOne(i)$ takes constant time, because it looks up record i in constant time, and checks the records to each side of it, and updates the pointers all in constant time
- $GetValue(i)$ just looks up record i and returns in value, so this also takes constant time.
- $GetBlockSize(i)$ takes O(n) in the worst case scenario, when $i = n$ and all bits 1 through n are connected.