1 Quercus Component

1.
   - `c) code`
   - `d) heap memory`
2.
   - `a) number of links to this inode`
   - `c) type of file system object`
   - `d) file size`
   - `f) array of pointers to blocks`
3.
   - `a) blcok bitmap`
   - `b) inode`
   - `c) data block`
4. `b) data block bitmap`

2 Markus Component

1. Algorithm A would complete faster.

   - We are only moving the used blocks to free blocks
   - It doesn't have to be concerned with finding related but scattered data blocks
     - Since it's not finding it, the total rotational delay time and seek time is low

   On the other hand Algorithm B has to find all data blocks scattered across the disk, and to make relevant data blocks contiguous, it has to move other blocks around. Finding each block adds seek and rotational delay time, and moving further adds the day. In total, the delay time is much larger than algorithm A.

2. Algorithm B would lead to better performance.

   The reason is that

   - For, algorithm A, the data blocks to a file are placed more randomly
     - the hard disk drive still has to move its arms to find all related data blocks
       - This adds rotational delay and seek time
   - For algorithm B, the data blocks to a file are placed contiguously
     - Then, we can write that the data blocks to a file are placed on current or nearby tracks.
     - Since data blocks are placed on current or nearby tracks, it requires little or no movement for disk arm
     - It follows from above that rotational delay and seek time are kept to minimum

3. **What changes, if any, are made to the inodes? Explain your answer.**

   - time modified, time accessed
     - time modified because data blocks are repositioned
     - time accessed because file has to be read in order to find its data blocks for defragmentation
   - the value of pointers to indirect blocks and data blocks
     - Is because under the defragementation algorithm, data blocks and indrect blocks are moved so they are located in contiguously. This changes block address. This change needs to be reflected in inode.

   What changes, if any, would need to be made to the directory metadata by the defragmentation algorithm? Explain your answer.

   - i-number in each directory entry
     - Is because inodes are being moved around from a group block to another so related directory inodes are put in group as close as possible
   - Pointer to data blocks
     - Is because defrag algorithm causes data blocks to be placed contiguously
     - Since directory inode stores addresses to these blocks, changes has to be made to reflect the changes
   - Time accessed
     - Is because directory to be accssed to find all its data blocks before moving around.
   - Time modified
     - Is because i-number in directory entries, and inode's address to directory blocks are changed

4. **What changes, if any, are made to the FAT (File Allocation Table) by the defragmentation algorithm? Explain your answer.**

   - pointers to next data block
     - FAT is a file system based on linked-list based allocation.
     - Linked list allocation uses next member variable to locate the next data block
     - Defrag moves the data blocks around, and this affects pointers to next data block and pointer to first data block in directory entry

   What changes, if any, are made to the directory metadata by the defragmentation algorithm? Explain your answer.

   - time accessed, time modified, i-number in each directory entry, pointer to first data block containing directory entries

- Because directory must be read to get it's data blocks, so they can be placed contiguously. Since file is read, this means file is accsssed. Since file is accssed, we can conclude time accsseed must be changed.
- Since defragmentation causes data blocks are being moved around. This affects i-number and pointer to first data block
- Since i-number and first pointer are changed, we can write directory is modified. Since directory is modified, we can conclude must be reflected in time modified