1. **Algorithms**:

a. **Search and Match Algorithms**: ATS uses algorithms to search and match keywords and phrases in resumes to job descriptions. These algorithms can be based on string matching techniques, such as the Knuth-Morris-Pratt algorithm or the Boyer-Moore algorithm, to find relevant candidate profiles.

b. **Sorting and Ranking Algorithms**: When sorting and ranking candidate profiles, ATS may use various algorithms, such as quicksort or mergesort, to efficiently organize the list of applicants based on factors like qualifications, experience, or relevance to the job.

c. **Recommendation Algorithms**: Some ATS systems employ recommendation algorithms to suggest suitable candidates for a given job posting. These algorithms can use collaborative filtering or content-based recommendation techniques to provide relevant candidate suggestions.

2. **Dynamic Programming**:

a. **Resume Parsing**: Dynamic programming can be used to parse and extract structured data from resumes. This involves breaking down the document into sections (e.g., education, work experience) and extracting relevant information, optimizing the process to handle various resume formats and languages.

b. **Optimal Workflow Sequencing**: ATS can employ dynamic programming to determine the optimal sequence of workflow steps in the recruitment process, taking into account various factors such as candidate availability, interview scheduling, and resource allocation.

3. **Optimal Memory Utilization**:

a. **Data Storage**: To efficiently utilize memory resources, ATS systems can employ data compression techniques and optimized data structures for storing candidate information, job postings, and other related data. This ensures that large volumes of data can be managed within available memory constraints.

b. **Caching**: ATS can utilize caching mechanisms to store frequently accessed data, such as job descriptions and common candidate qualifications, in memory. This reduces the need to retrieve data from the database repeatedly, improving system performance.

c. **Memory Management**: Effective memory management techniques, such as garbage collection and memory pooling, ensure that memory is used optimally, and resource leaks are minimized.