Online Algorithms –结构调整

In this section, we introduce the concept of online algorithms, which are algorithms that process data in a sequential manner. The difference between online and offline algorithms is that online algorithms must make decisions without knowing future inputs, while offline algorithms have access to the entire input data.

In the context of memory management, online algorithms are used to manage memory allocation and deallocation. They are particularly useful in scenarios where the input data is too large to fit into memory or when the size of the input data is not known in advance.

The goal of an online algorithm is to minimize the amount of memory used while maximizing the performance of the system. This is achieved by making decisions based on the available information at each step, rather than waiting for the entire input to be available.

There are several types of online algorithms, including paging, cache management, and linked lists. Each of these algorithms has its own set of challenges and trade-offs, and choosing the right algorithm depends on the specific requirements of the application.

In the next section, we will discuss the concept of competitive analysis, which is used to evaluate the performance of online algorithms. This will provide a framework for comparing different algorithms and determining their effectiveness.

In summary, online algorithms are a crucial component of modern computing systems, and understanding their concepts and techniques is essential for anyone working in the field of computer science.

בבלום בבלום

In this section, we will introduce the concept of online algorithms, which are algorithms that process data in a sequential manner. The difference between online and offline algorithms is that online algorithms must make decisions without knowing future inputs, while offline algorithms have access to the entire input data.

In the context of memory management, online algorithms are used to manage memory allocation and deallocation. They are particularly useful in scenarios where the input data is too large to fit into memory or when the size of the input data is not known in advance.

The goal of an online algorithm is to minimize the amount of memory used while maximizing the performance of the system. This is achieved by making decisions based on the available information at each step, rather than waiting for the entire input to be available.

There are several types of online algorithms, including paging, cache management, and linked lists. Each of these algorithms has its own set of challenges and trade-offs, and choosing the right algorithm depends on the specific requirements of the application.

In the next section, we will discuss the concept of competitive analysis, which is used to evaluate the performance of online algorithms. This will provide a framework for comparing different algorithms and determining their effectiveness.

In summary, online algorithms are a crucial component of modern computing systems, and understanding their concepts and techniques is essential for anyone working in the field of computer science.
OPT(σ) = \min\{B, n\}

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.

OPT(σ) = \min\{B, n\}.
\[
\theta(B - 1) = (1 - \theta) \left( \frac{B - 1}{B} \right)
\]

\[
B(\theta) = (1 - \theta)
\]

\[
\theta = \frac{1}{B + 1}
\]

\[
\text{makespan} = \max_j l_j^i
\]

\[
\text{OPT} = ON = 0, \, n = 0
\]

\[
\text{OPT} = \text{OPT}^{(n-1)} \leq \text{OPT}^{(n)}
\]

\[
\text{OPT}^{(n)} = \min_j l_j^{n-1} + a_n
\]

\[
\text{OPT}^{(n)} \geq \text{OPT}^{(n)} = \max_j l_j^n \geq \frac{1}{m} \sum_{j=1}^{m} l_j^n = \frac{1}{m} \sum_{i=1}^{m} a_i
\]

\[
\text{OPT}^{(n)} \leq \frac{1}{m} \sum_{i=1}^{n-1} a_i + a_n = \frac{1}{m} \sum_{i=1}^{m} a_i + (1 - \frac{1}{m}) a_n \leq \text{OPT}(\sigma^{n+1}) + (1 - \frac{1}{m}) \text{OPT}(\sigma^{(n)}) = (2 - \frac{1}{m}) \text{OPT}(\sigma^{(n)})
\]

\[
\text{OPT}^{(n)} = \min_j l_j^{n-1} + a_n
\]