Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Реферат

По дисциплине «Основы алгоритмизации и программирования»

На тему «Нотации оценки алгоритма»

 Выполнил:

Студент 1 курса 6 группы

Романов Игорь Вячеславович

Преподаватель: доц Белодед Н.И.

2024, Минск

**Содержание**

[Введение 3](#_Toc166240511)

[Аспекты 3](#_Toc166240512)

[Омега-большое (Ω-нотация) 3](#_Toc166240513)

[Тэта-большое (Θ-нотация) 4](#_Toc166240514)

[О-большое (O-нотация) 4](#_Toc166240515)

[Применение 4](#_Toc166240516)

[Заключение 5](#_Toc166240517)

Введение

Рациональное использование ресурсов, таких как память и время, является ключевым аспектом разработки программного обеспечения. Хороший программист стремится не только к написанию рабочего кода, но и к созданию эффективного решения, которое оптимально использует доступные ресурсы.

В контексте оценки сложности алгоритмов в зависимости от входных данных, нотация "О-большое" (Big O) играет важную роль. Она позволяет оценить, как быстро растет время выполнения или использование памяти алгоритма при увеличении размера входных данных. Это помогает программистам сделать обоснованный выбор алгоритмов в зависимости от требований проекта и ресурсов, которые им доступны.

В дополнение к большому О, существуют и другие нотации, такие как "Омега-нотация" и "Тета-нотация". В данном исследовании мы рассмотрим их характеристики и основные отличия.

Аспекты

При рассмотрении обозначения "O-большое" важно учитывать сложность алгоритма в зависимости от данных, таких как время выполнения кода и место его применения. Поэтому "O-большое" является лишь одним из способов классификации сложности. Сложность - это приблизительная мера эффективности алгоритма, связанная с каждым написанным алгоритмом.

Важно также определить три уровня (лучший случай, средний случай и худший случай), которые известны как асимптотические обозначения. Эти обозначения позволяют ответить на вопросы о том, не становится ли выполнение алгоритма медленным при увеличении размера входных данных и сохраняет ли он быстрое время выполнения при увеличении размера входных данных.

Омега-большое (Ω-нотация)

Омега-большое (обычно обозначаемая как Ω) представляет собой асимптотическое обозначение для наилучшего случая или минимальную скорость роста для данной функции. Это обозначение дает нам асимптотическую нижнюю границу скорости роста времени выполнения алгоритма.

Данная нотация не является лучшим решением для оценки алгоритма. Поскольку на практике хоть и достигаются идеальные условия для алгоритмов, но для качественной оценки работы кода и выявления затруднений необходимо учитывать более сложные случаи.

Тэта-большое (Θ-нотация)

Тета (обычно обозначаемая как Θ) является асимптотической записью для обозначения асимптотически жесткой границы скорости роста времени выполнения алгоритма. В такой нотации рассматривается и лучший, и худший случай.

О-большое (O-нотация)

O-большое (обычно обозначаемое как O) представляет собой асимптотическое обозначение для наихудшего случая или потолка роста для данной функции. Оно дает нам асимптотическую верхнюю границу скорости роста времени выполнения алгоритма.

Применение

На практике обычно выбирается наихудший сценарий (т.е. O-большое), так как что нельзя ожидать, что алгоритм будет запускаться в лучших или даже средних условиях. Это позволяет делать аналитические заявления, такие как производительности алгоритма.

Возьмем в пример пузырьковую сортировку массива. В своем лучшем случае, когда массив уже отсортирован, алгоритм будет составлять Ω(N). Однако, если массив будет не упорядочен, то сложность данного алгоритма составит уже O(N2), что значительно замедлит работу программы.

Другая ситуация обстоит с пирамидальной сортировкой. В любом случае программа будет перестраивать элементы массива в бинарную кучу. И тогда сложность алгоритма будет O(logn) = Θ (logn) = Ω(logn).

Поэтому необходимо при анализе тех или иных алгоритмов необходимо рассматривать все случае возможные случаи входных данных и прогнозировать дальнейшее поведение.

Заключение

Таким образом, в анализе сложности алгоритмов играет ключевую роль нотация "O-большое" (Big O), которая позволяет оценить время выполнения и использование памяти алгоритма в зависимости от размера входных данных. Это помогает программистам выбирать наиболее подходящие алгоритмы с учетом требований проекта и доступных ресурсов.

Однако помимо "O-большого" существуют и другие нотации, такие как "Омега-нотация" и "Тета-нотация", которые позволяют более детально анализировать сложность алгоритмов в различных сценариях. Они предоставляют информацию о лучшем, среднем и худшем случае выполнения алгоритма, что позволяет сделать более точные выводы о его производительности и эффективности.