

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н.И. ЛОБАЧЕВСКОГО
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МАТЕМАТИКИ И МЕХАНИКИ

Учебный курс

МЕТОДЫ ПРОГРАММИРОВАНИЯ - 2





Нижегородский государственный университет им. Н.И. Лобачевского
Институт информационных технологий, математики и механики

Учебный курс:

Методы программирования - 2

Тема 3:

Динамические структуры данных

Гергель В.П., профессор ,
директор института ИТММ

Глава 1.

Структура действия и структуры данных

1.3. Динамические структуры данных

1. Преобразование структур данных в процессе вычислений
2. Понятие динамической структуры. Примеры

1.4. Динамические структуры и структуры хранения

1. Общая структура хранения элементов и программы реализации отношения включения.

Практическая работа 3: Реализация стека

2. Сравнение линейных и динамических структур
3. Проблема эффективного использования памяти.

Практическая работа 4: Реализация очереди

4. Использование нескольких структур и необходимость динамического распределения памяти. Задача реализации двух стеков

Вопросы для обсуждения

1.3. Динамические структуры данных...

1. Преобразование структур данных в процессе вычислений...

- ☑ Структуры данных являются операндами операций обработки
- ☑ Результаты вычислений также являются структурами, модель которых может как совпадать, так и отличаться от структуры исходных данных

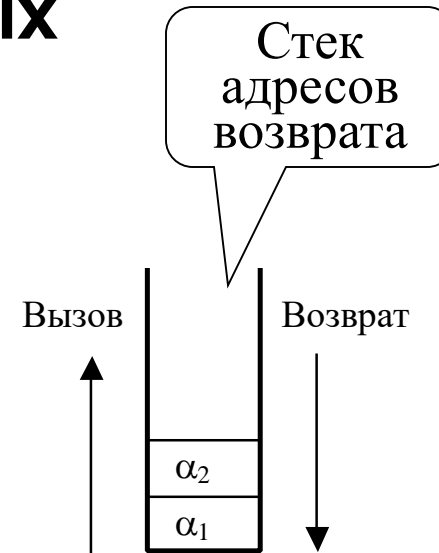
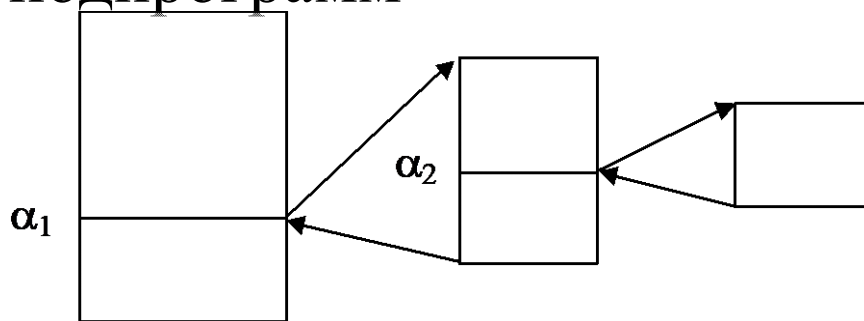
Пример: Произведение векторов

$$ab^T = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} * (b_1 \dots b_m) = \begin{pmatrix} c_{11} \dots c_{1m} \\ \dots \\ c_{n1} \dots c_{nm} \end{pmatrix} = c$$

1.3. Динамические структуры данных...

1. Преобразование структур данных в процессе вычислений...

Пример 1.4. Организация последовательного вызова подпрограмм



- ☑ Для возврата в точку вызова необходимо запоминать *адрес возврата*
- ☑ При завершении вызываемой подпрограммы для возврата используется последний запомненный адрес
- ☑ В ходе вызова подпрограмм количество запоминаемых адресов постоянно изменяется (увеличивается при вызове очередной подпрограммы и уменьшается после завершения работы текущей подпрограммы)

1.3. Динамические структуры данных...

1. Преобразование структур данных в процессе вычислений...

☑ Отличительная особенность – структура исходных и результирующих данных являются близкими

Выполним анализ рассмотренного примера

- Текущий набор адресов – линейная структура

$$S_n = (a_1 a_2 \dots a_n)$$

- Пусть T – операция исключения последнего адреса. Тогда

$$T(a_1 a_2 \dots a_n a_{n+1}) = (a_1 a_2 \dots a_n)$$

- Пусть P – операция добавления нового адреса. Тогда

$$P(a_{n+1}; (a_1 a_2 \dots a_n)) = (a_1 a_2 \dots a_n a_{n+1})$$

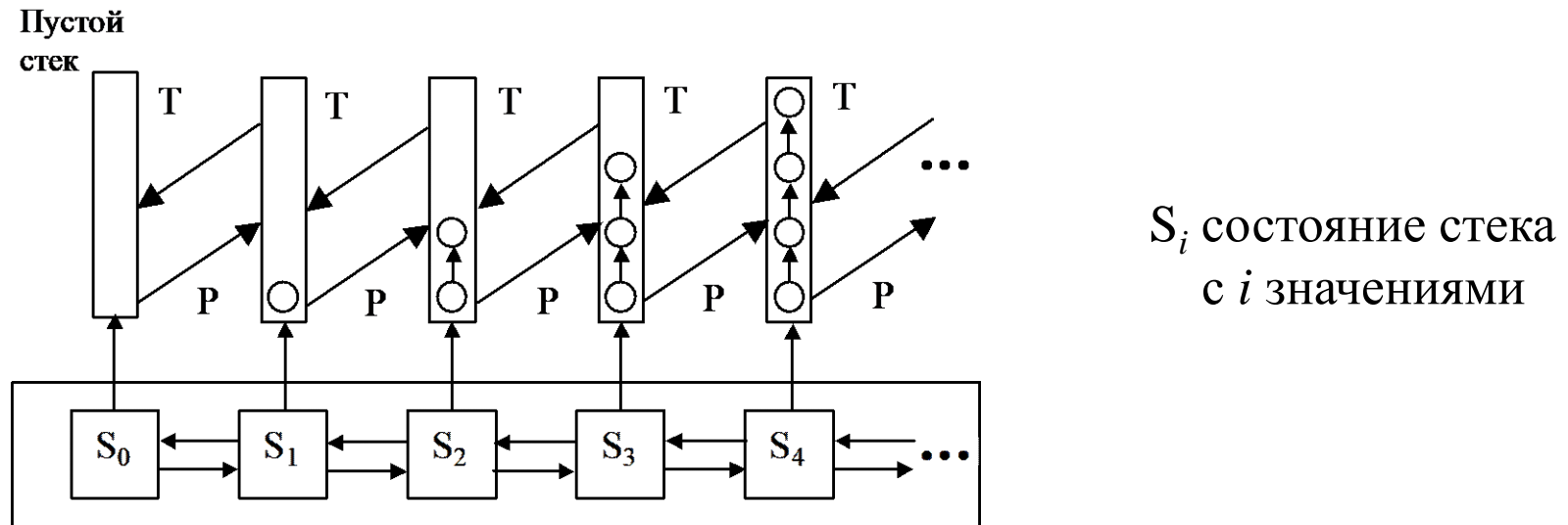
☑ Орграф результата является подорграфом орграфа операнда или включает его



1.3. Динамические структуры данных...

1. Преобразование структур данных в процессе вычислений...

- ✓ Последовательное применение операций T и P позволяет получить набор состояний стека адресов



- ✓ Мы имеем множество M элементов, у которых есть имена и значения

- ✓ $S_i \subset S_{i+1}$, т.е. элементы связаны отношением включения подграфов

1.3. Динамические структуры данных...

1. Преобразование структур данных в процессе вычислений

Пусть:

- \mathcal{P}_1 - отношение следования, порождаемое операцией вставки,
- \mathcal{P}_2 - отношение следования, порождаемое операцией исключения.

Тогда *стек* есть структура

$$S = (\mathcal{M}, \mathcal{P}_1, \mathcal{P}_2),$$

в которой

- каждый элемент – структура,
- в любой момент существует только один конкретный элемент из \mathcal{M} ,
- элементы частично упорядочены по включению.

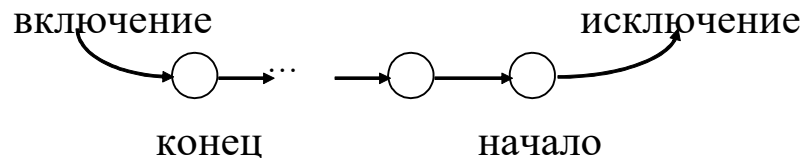


1.3. Динамические структуры данных

2. Понятие динамической структуры

Определение 1.9. *Динамическая структура* есть математическая структура, которой соответствует частично-упорядоченное (по включению) базовое множество M , элементы которого являются структурами данных. При этом отношения включения индуцируются операциями преобразования структуры данных.

Пример 1.5. **Очередь** (вставка в конец очереди, исключение из начала – дисциплина FIFO – first in, first out)



Пример 1.6. **Дек** (dequeue – double ended queue) – вставка и исключение для начала и конца дека – дисциплина FOLIFOLO – first or last in, first or last out)



1.4. Динамические структуры и структуры хранения...

1. Общая структура хранения элементов и программы реализации отношения включения

- ☑ В каждый текущий момент времени в памяти хранится только один элемент базисного множества
- ☑ Для преобразования текущего элемента к следующему должны быть разработаны *программы поддержки* динамической структуры
- ☑ Структура хранения текущего элемента должна иметь достаточно общий характер для хранения любых элементов базисного множества структуры данных элемента

1.4. Динамические структуры и структуры хранения...

1. Пример разработки структуры хранения

Практическая работа 3: Реализация стека



1.4. Динамические структуры и структуры хранения...

2. Сравнение линейных и динамических структур

	Линейные структуры	Динамические структуры
1	Базисное множество – множество элементов	Элементы базисного множества являются структурами ("ДС – структуры структур")
2	Базисное отношение – отношение следования	Базисное отношение – отношение включения
3	В структуре хранения хранятся все элементы структуры	В структуре хранения хранится только текущий элемент структуры
4	Отношение следования реализуется при помощи адресной арифметики	Отношение включения реализуется при помощи программ

Определение 1.10. Программы, реализующие отношение включения, называются *средствами поддержания динамической структуры*



1.4. Динамические структуры и структуры хранения...

3. Проблема эффективного использования памяти...

- ☑ При использовании стека может возникнуть ситуация исчерпания памяти для хранения значений – *ограничение реализации* теоретически неограниченного стека
- ☑ Невозможность использования стека из-за переполнения возникает только в момент полного использования всей выделенной для стека памяти

Определение 1.11. Для оценки эффективности использования памяти введем показатель E_{mem} , определяемый как отношение объема используемой памяти к общему размеру выделенной памяти.



1.4. Динамические структуры и структуры хранения...

3. Проблема эффективного использования памяти

Практическая работа 4: Реализация очереди



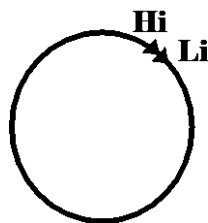
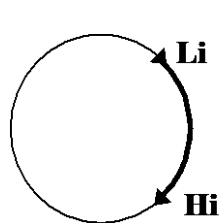
1.4. Динамические структуры и структуры хранения...

4. Использование нескольких структур и необходимость динамического распределения памяти

Пример 1.7. Стек для хранения фиксированного количества последних записанных значений

Стек подобного вида может быть использован, например, для запоминания данных, достаточных для восстановления текущего состояния программы перед выполнением очередной операции обработки (для реализации процедуры типа операции отмены в редакторе текстов). При реализации данного механизма обычно фиксируется максимальная глубина отката; при заполнении памяти из стека удаётся наиболее старая записанная информация.

↪ Наиболее простой способ реализации такого стека – использование кольцевого буфера (при заполнении стека следующий для использования по кольцу элемент памяти содержит подлежащее удалению значение данных)



Как реализовать процедуру отмены операции отката ?

1.4. Динамические структуры и структуры хранения...

4. Использование нескольких структур и необходимость динамического распределения памяти...

Пример 1.8. Задача использования двух стеков

☑ При использовании двух стеков может возникнуть ситуация переполнения одного стека, в то время как в другом стеке свободная память может еще присутствовать – эффективность использования памяти в этом случае не будет являться максимально-возможной

↪ Возможное решение проблемы – использование общей памяти для этих стеков, в которой один стек будет "расти" слева направо, второй стек – справа налево.



☑ При таком подходе свободная память является общей и ситуация переполнения стеков возникает только при полном исчерпании памяти

1.4. Динамические структуры и структуры хранения

4. Использование нескольких структур и необходимость динамического распределения памяти

Определение 1.13. Распределение памяти до начала процесса вычислений называется *статическим*.

Распределение памяти в ходе выполнения программы называется *динамическим распределением памяти*.

- ☑ Как распределить память для большего, чем 2, количества стеков ?

Заключение

- Понятие динамической структуры данных
- Необходимость разработки общей структуры хранения для элементов базисного множества
- Необходимость программной реализации отношения включения
- Отличительные особенности линейных и динамических структур
- Структура хранения в виде кольцевого буфера
- Статическое и динамическое распределение памяти
- Примеры реализации динамических структур: стеки и очереди
- Программирование с защитой от ошибок

Вопросы для обсуждения

- Классификация видов структур данных: линейные, динамические...
- Проблема разных типов значений для элементов базисных множеств
- Принципы программирования с защитой от ошибок (накладные расходы, увеличение объема программирования, существующие подходы...)
- Достоинства и недостатки динамического распределения памяти

Темы заданий для самостоятельной работы

- Реализация стеков и очередей с использованием шаблонов
- Разработка программы для вычисления арифметического выражения в постфиксной записи

Следующая тема

- Динамическое распределение памяти

Контакты

Нижегородский государственный университет
им. Н.И. Лобачевского (www.unn.ru)

Институт информационных технологий, математики
и механики (www.itmm.unn.ru)

603950, Нижний Новгород, пр. Гагарина, 23,
р.т.: (831) 462-33-56,

Гергель Виктор Павлович

(<http://www.software.unn.ru/?dir=17>)

E-mail: gergel@unn.ru

