**работа № 4. ИССЛЕДОВАНИЕ МЕТОДОВ ПОИСКА**

***Цель работы*:** Изучить основные методы быстрого поиска; получить практические навыки программирования задач быстрого поиска.

Варианты заданий

1. Исследовать алгоритмы поиска (успешный и безуспешный поиск) в соответствии с вариантами:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № варианта | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 1 | **+** | **+** | **+** | **+** |  |  |  | **+** | **+** |  |  |  | **+** | **+** |
| 2 | **+** | **+** | **+** |  | **+** |  |  | **+** |  | **+** |  |  | **+** | **+** |
| 3 | **+** | **+** | **+** |  |  | **+** |  | **+** |  |  | **+** |  | **+** | **+** |
| 4 | **+** | **+** | **+** |  |  |  | **+** | **+** |  |  |  | **+** | **+** | **+** |
| 5 | **+** | **+** |  | **+** | **+** |  |  | **+** | **+** |  |  |  | **+** | **+** |
| 6 | **+** | **+** |  | **+** |  | **+** |  | **+** |  | **+** |  |  | **+** | **+** |
| 7 | **+** | **+** |  | **+** |  |  | **+** | **+** |  |  | **+** |  | **+** | **+** |
| 8 | **+** | **+** |  |  | **+** | **+** |  | **+** |  |  |  | **+** | **+** | **+** |
| 9 | **+** | **+** |  |  | **+** |  | **+** | **+** | **+** |  |  |  | **+** | **+** |
| 10 | **+** | **+** |  |  |  | **+** | **+** | **+** |  | **+** |  |  | **+** | **+** |

Номер варианта *N* определяется по следующей формуле (*i* – номер студента по списку группы, *k* – число вариантов (10), **mod** – остаток целочисленного деления)

*N* = (*i* – 1) **mod** *k* + 1.

Номера столбцов соответствуют следующим алгоритмам поиска:

1. Последовательный поиск в упорядоченной таблице.

2. Обычный бинарный поиск.

3. Однородный бинарный поиск с вычислением значений δ.

4. Однородный бинарный поиск с дополнительной таблицей с вычисленными заранее значениями δ.

5. Поиск Фибоначчи с проверкой условия *i* ≤ 0.

6. Поиск Фибоначчи без проверки условия *i* ≤ 0.

7. Интерполяционный поиск.

8. Поиск в обычном дереве бинарного поиска (без балансировки).

9. Поиск в АВЛ-дереве (узел не содержит поле для указателя на отца).

10. Поиск в RB-дереве (узел не содержит поле для указателя на отца).

11. Поиск в АВЛ-дереве (в узле есть поле *Father* для указателя на отца).

12. Поиск в RB-дереве (в узле есть поле *Father* для указателя на отца).

13. Цифровой поиск.

14. Поиск с хешированием.

2. Исследовать алгоритмы включения и исключения для динамических таблиц, представленных с помощью деревьев бинарного поиска с балансировкой (АВЛ- или RB-дерево в соответствии с вариантом из предыдущего пункта).

Подготовка к работе

1. Разработать алгоритмы и программы проведения экспериментальных исследований методов поиска (как успешный, так и безуспешный поиск).

2. Разработать алгоритмы и программы проведения экспериментальных исследований операций с динамическими таблицами (включение и исключение).

3. Разработать алгоритмы и программы для методов поиска в соответствии с заданным вариантом.

4. Подготовить таблицы для записи результатов экспериментальных исследований. Лучше всего предусмотреть заполнение этих таблиц в программе проведения экспериментальных исследований. Эти таблицы представляются двумерным массивом, содержимое которого легко можно отобразить на экране монитора после завершения экспериментов.

Выполнение работы

1. Исследовать алгоритмы поиска (успешный и безуспешный поиск).

Для этого вначале формируется исходная упорядоченная таблица из *N* имен, представляющих собой четные положительные числа в пределах от 2 до 2*N*, и представляется соответствующей структурой данных (массив, дерево). Значение величины *N* выбирается исходя из соображений возможности фиксирования времени вычислений (*по возможности*). Время, затрачиваемое на заполнение исходной таблицы, при исследовании алгоритмов не учитывается.

Исследование методов поиска состоит в проведении *K* (*K* ≥ 1000) статистических испытаний. Каждое испытание включает в себя генерацию случайного ключа поиска, прогон для этого ключа всех исследуемых алгоритмов (программ), подсчитывая при этом число сравнений имен и фиксируя (*по возможности*) время поиска. Полученные результаты суммируются с соответствующими значениями, вычисленными на предыдущих испытаниях, т. е. идет накопление значений. После проведения всех *K* испытаний деление полученных значений на *K* дает средние значения соответствующих величин, которые заносятся в таблицу результатов.

Такие исследования проводятся для различных значений *N* (число размеров таблиц ≥ 5). По полученным данным необходимо построить аппроксимирующие функции и вычислить аналитические зависимости времени поиска от *N*.

Для исследования успешного поиска генерируются случайные четные ключи в диапазоне от 2 до 2*N*, а безуспешного поиска – нечетные ключи в диапазоне от 1 до 2*N* + 1.

2. Исследовать алгоритмы включения и исключения для динамических таблиц.

Для этого вначале создается либо пустое дерево поиска, либо можно воспользоваться деревом, построенным в предыдущем пункте (содержит только четные ключи).

Исследование состоит в проведении *K* (*K* ≥ 1000) статистических испытаний. Каждое испытание включает в себя генерацию случайного запроса (ключа поиска) в диапазоне от 1 до *N*. Обработка запроса заключается в следующем: если ключ уже есть в дереве поиска, то соответствующий узел исключается из дерева, в противном случае – выполняется операция включения нового узла. При этом накапливаются значения следующих данных: число операций вращения при балансировке (для АВЛ-деревьев двойное вращение считать как два вращения) и время (*по возможности*) обработки запроса. После проведения всех *K* испытаний деление полученных значений на *K* дает средние значения соответствующих величин, которые заносятся в таблицу результатов.

Рекомендуется также в качестве результата исследования определить максимальное число вращений, потребовавшихся для балансировки дерева при проведении этих *K* испытаний.

Такие исследования проводятся для различных значений *N* (число размеров таблиц ≥ 5).

3. Продемонстрировать работу программ преподавателю.

Содержание отчета

1. Цель работы.

2. Информация в соответствии с подготовкой к работе.

3. Скриншот с результатами экспериментальных исследований алгоритмов.

4. Выводы.