**Приложение №1**

|  |  |
| --- | --- |
| **Дата** | 08.12.2023 |
| **Тема** | Цели и задачи практики, содержание практики. Основные темы и варианты заданий. Инструктаж по технике безопасности и пожарной безопасности. Материалы, необходимые для работы. Требования к отчету по практике. Разработка и реализация алгоритма пирамидальной сортировки. Понятие сортировки в С#. Виды сортировки. Механизм пирамидальной сортировки. |
| **Задания** | 1. Ознакомится с целями и задачами  2. Прослушать инструктаж по технике безопасности  3. Ознакомится с основными темами и вариантами заданий  4. Ознакомится с требованиями к отчету по практике  5. Основные понятия пирамидальной сортировки  6. Ознакомление с устройством и логикой работы пирамидальной сортировки |

1. **Ознакомится с целями и задачами:**

Цели практики: приобретение начальных практических навыков по разработке программных модулей. Задачи практики:

- получение практических навыков по разработке спецификации отдельных модулей программного продукта;

- получение практических навыков по разработке программного кода модуля программного продукта на основе спецификации.

**2. Прослушать инструктаж по технике безопасности:**

1.1 Настоящая инструкция по охране труда программиста, занятого эксплуатацией персональных электронно-вычислительных машин (ПЭВМ) и видеодисплейных терминалов (ВДТ), разработана с учетом условий его работы в конкретной организации. 1.2. На программиста могут воздействовать опасные и вредные производственные факторы: А) физические: - повышенные уровни электромагнитного излучения; - повышенные уровни рентгеновского излучения; - повышенные уровни ультрафиолетового излучения; - повышенный уровень инфракрасного излучения; - повышенный уровень статического электричества; - повышенные уровни запыленности воздуха рабочей зоны; - повышенное содержание положительных аэроионов в воздухе рабочей зоны; - пониженное содержание отрицательных аэроионов в воздухе рабочей зоны; - пониженная или повышенная влажность воздуха рабочей зоны; - пониженная или повышенная подвижность воздуха рабочей зоны; - повышенный уровень шума; - повышенный или пониженный уровень освещенности; - повышенный уровень прямой блесткости; - повышенный уровень отраженной блесткости; - повышенный уровень ослепленности; - неравномерность распределения яркости в поле зрения; - повышенная яркость светового изображения; - повышенный уровень пульсации светового потока; - повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека; Б) химические: - повышенное содержание в воздухе рабочей зоны двуокиси углерода, озона, аммиака, фенола, формальдегида и полихлорированных бифенилов; B) психофизиологические: - напряжение зрения; - напряжение внимания; - интеллектуальные нагрузки; - эмоциональные нагрузки; - длительные статические нагрузки; - монотонность труда; - большой объем информации, обрабатываемой в единицу времени; - нерациональная организация рабочего места; Г) биологические: - повышенное содержание в воздухе рабочей зоны микроорганизмов. 1.3. К работам программистом допускаются: - лица не моложе 18 лет, прошедшие обязательный при приеме на работу и ежегодные медицинские освидетельствования на предмет пригодности для работы; - прошедшие вводный инструктаж по охране труда; - прошедшие обучение безопасным приемам и методам труда по программе, утвержденной руководителем предприятия (работодателем), разработанной на основе Типовой программы, и прошедшие проверку знаний, в том числе по электробезопасности; - прошедшие курс обучения на персональном компьютере с использованием конкретного программного обеспечения; - прошедшие инструктаж по охране труда на конкретном рабочем месте по данной инструкции. 1.4. Программист должен быть обеспечен СИЗ в соответствии с Межотраслевыми правилами обеспечения работников специальной одеждой, специальной обувью и другими средствами индивидуальной защиты, утвержденными Приказом Минздравсоцразвития России от 01.06.2009 N 290н; выдаваемые работникам средства индивидуальной защиты должны соответствовать характеру и условиям работы и обеспечивать безопасность труда. Не допускаются приобретение и выдача работникам средств индивидуальной защиты без сертификата соответствия. Характеристика выданных СИЗ (номенклатура, срок выдачи и нормы соответствия) устанавливается из личных карточек работников, занятых на определенном рабочем месте, Нормативные номенклатура и сроки выдачи СИЗ определяются согласно Типовым отраслевым нормам бесплатной выдачи рабочим и служащим специальной одежды, специальной обуви и других **СИЗ**.

**2. ТРЕБОВАНИЯ БЕЗОПАСНОСТИ ПЕРЕД НАЧАЛОМ РАБОТЫ**

**2.1.**  Перед началом работы программист обязан: -- осмотреть и привести в порядок рабочее место; ­-- отрегулировать освещённость на рабочем месте, убедится в достаточности освещённости, отсутствии отражений на экране, отсутствии встречного светового потока; -- проверить правильность подключения оборудования в электросеть; -- протереть специальной салфеткой поверхность экрана; -- убедится в отсутствии дискет в дисководах процессора персонального компьютера; -- проверить правильность установки стола, стула, подставки дляног, пюпитра, положения оборудования, угла наклона экрана, положение клавиатуры и, при необходимости, произвести регулировку рабочего стола и кресла, а также расположение элементов компьютера в соответствии с требованиями эргономики и в целях исключения неудобных поз и длительных напряжений тела.

**2.2** При включении компьютера соблюдать правила электробезопасности.

**2.3** Программисту запрещается приступать к работе при: отсутствии на ВДТ гигиенического сертификата, включающего оценку визуальных параметров; - отсутствии информации о результатах аттестации условий труда на данном рабочем месте или при наличии информации о несоответствии параметров данного оборудования требованиям санитарных норм; - отсутствии защитного экранного фильтра класса "полная защита"; отключенном заземляющем проводнике защитного фильтра; - обнаружении неисправности оборудования; - отсутствии защитного заземления устройств ПЭВМ и ВДТ; - отсутствии углекислотного или порошкового огнетушителя и аптечки первой помощи; - нарушении гигиенических норм размещения ВДТ (при однорядном расположении менее 1 м от стен, при расположении рабочих мест в колонну на расстоянии менее 1,5 м, при размещении на площади менее 6 кв. м на одно рабочее место, при рядном размещении дисплеев экранами друг к другу).

**3. ТРЕБОВАНИЯ БЕЗОПАСНОСТИ ВО ВРЕМЯ РАБОТЫ**

**3.1** Программист во время работы обязан: - выполнять только ту работу, которая ему была поручена и по которой он был проинструктирован; - в течение всего рабочего дня содержать в порядке и чистоте рабочее место; - держать открытыми все вентиляционные отверстия устройств; - при необходимости прекращения работы на некоторое время корректно закрыть все активные задачи; - выполнять санитарные нормы и соблюдать режимы работы и отдыха; - соблюдать правила эксплуатации вычислительной техники в соответствии с инструкциями по эксплуатации; - соблюдать установленные режимом рабочего времени регламентированные перерывы в работе и выполнять в физкультпаузах и физкультминутках рекомендованные упражнения для глаз, шеи, рук, туловища, ног; - соблюдать расстояние от глаз до экрана в пределах 60 - 80 см.

**3.2** Программисту во время работы запрещается: прикасаться к задней панели системного блока (процессора) при включенном питании; переключать разъемы интерфейсных кабелей периферийных устройств при включенном питании; загромождать верхние панели устройств бумагами и посторонними предметами; допускать захламленность рабочего места бумагой - в целях недопущения накапливания органической пыли; производить отключение питания во время выполнения активной задачи; производить частые переключения питания; допускать попадание влаги на поверхность системного блока (процессора), монитора, рабочую поверхность клавиатуры, дисководов, принтеров и др. устройств; включать сильно охлажденное (принесенное с улицы в зимнее время) оборудование; производить самостоятельно вскрытие и ремонт оборудования.

**4. ТРЕБОВАНИЯ БЕЗОПАСНОСТИ В АВАРИЙНЫХ СИТУАЦИЯХ**

**4.1** Программист обязан: - во всех случаях обнаружения обрыва проводов питания, неисправности заземления и других повреждений электрооборудования, появления запаха гари немедленно отключить питание и сообщить об аварийной ситуации руководителю и дежурному электрику; - при обнаружении человека, попавшего под напряжение, немедленно освободить его от действия тока путем отключения электропитания и до прибытия врача оказать потерпевшему первую медицинскую помощь; - при любых случаях сбоя в работе технического оборудования или программного обеспечения немедленно вызвать представителя инженерно- технической службы эксплуатации вычислительной техники; - в случае появления рези в глазах, при резком ухудшении видимости - невозможности сфокусировать взгляд или навести его на резкость, появлении боли в пальцах и кистях рук, усилении сердцебиения немедленно покинуть рабочее место, сообщить о происшедшем руководителю работ и обратиться к врачу; - при возгорании оборудования отключить питание и принять меры к тушению очага пожара при помощи углекислотного или порошкового огнетушителя, вызвать пожарную команду и сообщить о происшествии руководителю работ.

**5. ТРЕБОВАНИЯ БЕЗОПАСНОСТИ ПОСЛЕ ОКОНЧАНИЯ РАБОТЫ**

**5.1** По окончании работ программист обязан соблюдать следующую последовательность выключения вычислительной техники: - произвести закрытие всех активных задач; - выполнить парковку считывающей головки жесткого диска (если не предусмотрена автоматическая парковка головки); - убедиться, что в дисководах нет дискет; - выключить питание системного блока (процессора); - выключить питание всех периферийных устройств; - отключить блок питания.

**5.2** По окончании работ программист обязан осмотреть и привести в порядок рабочее

место, повесить халат в шкаф и вымыть с мылом руки и лицо.

**3. Ознакомится с основными темами и вариантами заданий**

Ознакомился с основными темпами и вариантами заданий.

**4. Ознакомится с требованиями к отчету по практике:**

Дневник по практике должен содержать:

1.a титульный лист

1.b маршрутный лист

1.c график работы

1.d индивидуальный план

1.e рабочая программа

1.f ежедневные отчёты

1.g приложения

**5. Основные понятия пирамидальной сортировки**

Пирамидальная сортировка в некотором роде является модификацией такого подхода, как сортировка выбором, с тем лишь отличием, что минимальный(или максимальный) элемент из неотсортированной последовательности выбирается не за O(n) операций, а за O(log n). Соответственно и производительность такого метода будет не ~ O(n2), а O(n\*log n), т.е. наиболее быстрая для сортировки.

Естественно, без дополнительных манипуляций с неотсортированной последовательностью мы не добьемся выбора из нее максимального(в пирамидальной сортировке выбирается максимальный, поэтому дальше будем это всегда предполагать) элемента за O(log n). Точнее, для такого быстрого выбора из этой неотсортированной последовательности строится некоторая структура. И вот именно суть метода пирамидальной сортировки и состоит в построении такой структуры, называемой соответственно "пирамидой".

**Суть метода**

Теперь весь метод в целом. Вначале имеем исходную нашу неотсортированную последовательность. Из нее строится вышеупомянутая структура данных - пирамида. Пирамида обладает тем свойством, что в ее вершине - находится максимальный элемент. Кроме того, вершина — это элемент в структуре пирамиды с начальным индексом. Поэтому чтобы выбрать из пирамиды максимальный элемент - не нужно делать вообще ничего - нужно просто взять этот начальный элемент структуры пирамиды.

Пирамида построена, далее можно начинать основные итерации сортировки. На каждой итерации:

1. Вынимаем вершину пирамиды
2. На ее место вставляем последний в пирамиде элемент
3. "Просеиваем" этот элемент сквозь пирамиду.

И так до последней итерации, на которой из всей пирамиды останется только вершина, которую мы и вставим в конец нашей полученной отсортированной последовательности.

**6. Ознакомление с устройством и логикой работы пирамидальной сортировки**

**Структура пирамиды**

Теперь рассмотрим наконец, что же представляет из себя эта пирамида. Это бинарное дерево, в котором каждый элемент меньше либо равен его родителю.

Например, имеем исходную последовательность:

x = [22, 100, 44, 15, 2, 36, 53, 23, 82, 5]

Хранить пирамиду удобно в виде массиве. Нумерацию элементов пирамиды определяем просто: проходимся сверху вниз, слева направо по каждому элементу и нумеруем его. В итоге получим то, что изображено на рисунке(цифры черным цветом - искомые индексы).

Т. получим x[0] - вершина пирамиды, а левый и правый потомки каждого элемента x[i] будут соответственно: x[2\*i+1] и x[2\*i+2] и основное свойство пирамиды тогда можно записать как:

x[i] >= max( x[2\*i+1], x[2\*i+2] )

**Построение пирамиды**

Исходный массив делится пополам, вторая его половина уже принимается за пирамиду. Затем последовательно берутся элементы из первой половины и добавляются в пирамиду.

Может возникнуть вопрос: с чего это вдруг мы сразу полмассива считаем за пирамиду, а не строим ее шаг за шагом начиная с последнего элемента, добавляя туда потом предпоследний и т.д. до первого элемента? Дело все в том, что для второй половины нашего исходного массива основное свойство пирамиды - выполняется автоматически. Вернее, оно не нарушается, поскольку для элементов второй половины просто уже не будет потомков!!! Действительно: потомки для всех x[i], где i = n/2..n будут соответственно x[n+1]...x[2\*n+2], т.е. такие, которых в нашем массиве уже нет. А поэтому нет смысла для элементов второй половины строить пирамиду последовательно добавляя каждый элемент, т.к. в алгоритме текущий добавляемый элемент просто будет не с чем сравнивать, сыновей-то с такими индексами - просто нет!

Так что спокойно принимаем вторую половину нашей последовательности как пирамиду и приступаем к следующему этапу построения - добавлению элементов.

Чтобы добавить элемент в пирамиду - нужно поменять его с наибольшим ребенком, если последний превосходит добавляемый элемент. Затем тоже самое по отношению уже к новым его детям. Т.о. чтобы добавить в пирамиду новый элемент x[i], нужно:

* Добавить элемент x[i] слева к массиву уже имеющейся пирамиды
* Найти максимального из его сыновей: maxChild = max( x[2\*i+1], x[2\*i+2] )
* Если x[i] >= maxChild - завершение процедуры, элемент занимает положенное ему место, добавление завершено, иначе - шаг 4.
* maxChild > x[i] - поменять их местами(т.е. поменять местами x[i] и его большего ребенка).
* Перейти к шагу 2, учитывая новое положение добавляемого элемента.

При этом, как можно догадаться - просеивание, о котором велась речь выше — это тоже добавление, поскольку после удаления максимального элемента, т.е. вершины пирамиды, мы ставим на его место последний элемент, а это посути - опять же добавление элемента в пирамиду. Поэтому для просеивания нам не нужно писать еще одну функцию, мы просто вызываем добавление по отношению к этому новому элементу.

**Выводы:** Ознакомился с целями и задачами. Послушал инструктаж по технике безопасности. Ознакомился с основными темами и вариантами заданий. Подобрал материалы, необходимые для работы. Ознакомился с требованиями к отчету по практике. Составил индивидуальный план. Разработал и реализовал алгоритм пирамидальной сортировки.

Студент: / Ульянов Никита Анатольевич