实验报告

## 实验内容 : 内部排序算法的比较

## 需求分析

1.题目描述：

排序是数据结构中重要的一个部分，也是在实际开发中易遇到的问题，所以研究各种排算法的时间消耗对于在实际应用当中很有必要通过分析实际结合算法的特性进行选择和使

用哪种算法可以使实际问题得到更好更充分的解决！该系统通过对各种内部排序算法如直接插入排序，冒泡排序，简单选择排序，快速排序，希尔排序，堆排序、二路归并排序等，以关键码的比较次数和移动次数分析其特点，并进行比较,估算每种算法的时间消耗，从而比较各种算法的优劣和使用情况！

2,简要分析：

首先需要对直接插入排序，冒泡排序，简单选择排序，快速排序，希尔排序，堆排序等做出更改，添加计数比较次数和移动次数的估计量，mo以及 bi ．

通过比较前后的变化进行比较．

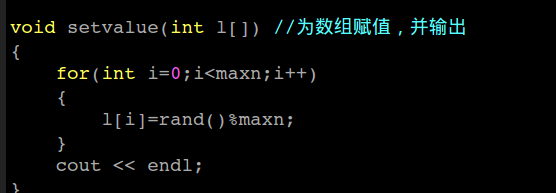
由于需要五次比较，那么就生成５次随机数数组，然后用不同的算法比较．

## 概要设计

首先需要实现６大排序算法，然后用rand()来生成随机数组，分别用６种不同的排序算法进行比较即可．

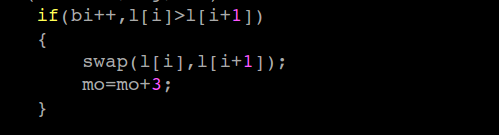
## 详细设计.

1. 首先，需要有一个生成数组的函数，已经实现如下：



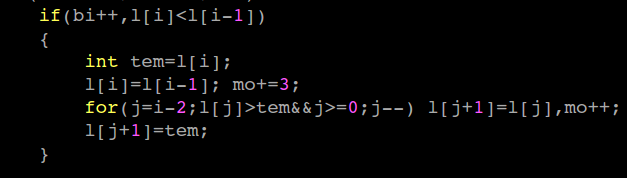
1. 分别对６大排序算法进行改造，在适当位置添加计数器，如下：

冒泡排序修改如下：

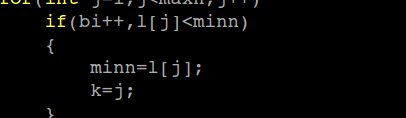


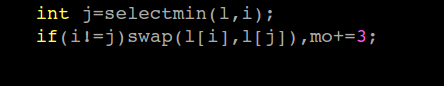
由于逗号表达式的真值在最后一位，那么，就可以子啊比较之前添加一个计数器即可：

直接插入排序如下 ;

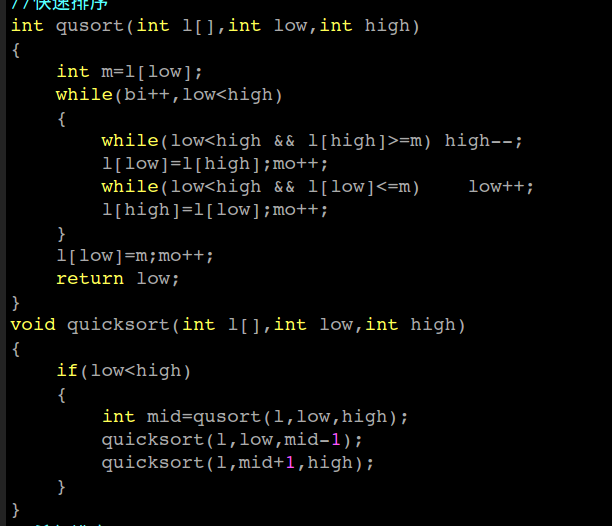


简单选择排序如下:

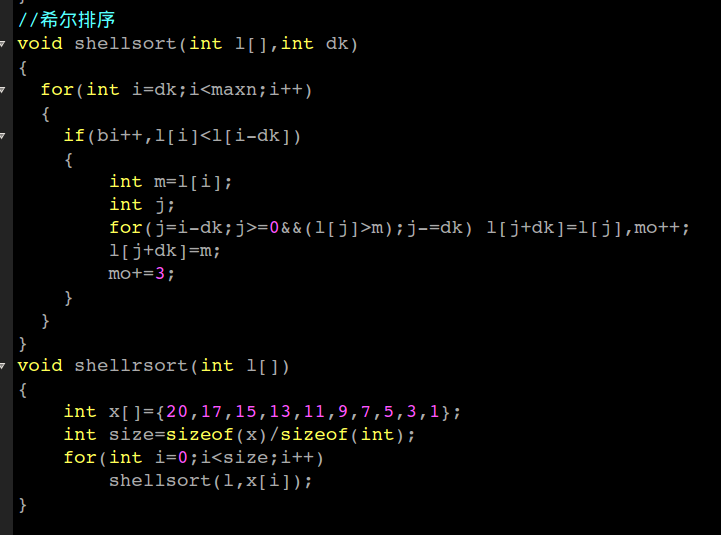




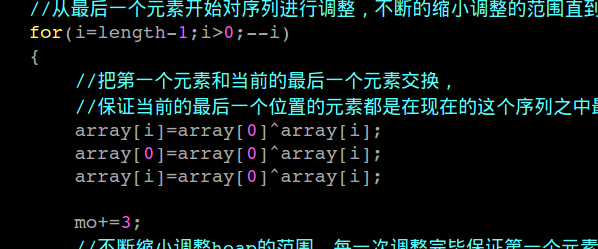
快速排序如下 ;



希尔排序如下:



堆排序如下:



## 用户手册

由于程序的输出较长,所有输出到了out.txt.由于在不同的机器上的运行结果不一样,这里就在我的机器上的运行结果:

生成的数组为 : 55 185 1 128 156 194 174 150 57 116 196 68 107 127 113 22 33 65 64 118 143 193 35 59 83 174 20 35 10 53 27 65 190 180 145 146 174 119 48 183 187 45 52 94 124 165 69 109 182 133 27 126 79 62 185 162 188 157 149 150 11 128 167 153 108 112 52 34 183 100 18 171 97 70 17 173 187 86 82 170 172 109 48 51 123 33 165 64 143 67 14 154 195 182 59 104 46 111 138 30 164 156 1 13 178 170 187 118 57 69 40 29 131 88 32 6 73 197 70 16 64 37 122 60 171 182 116 17 93 54 199 9 11 152 23 141 123 10 59 180 31 51 161 114 139 145 121 13 142 143 29 159 180 152 171 151 86 87 121 131 93 120 141 56 73 164 198 196 126 9 128 109 61 41 24 0 186 97 13 80 192 195 191 173 99 162 76 185 1 149 116 95 70 57 151 95 173 101 43 51

冒泡排序:

移动次数　：30231 比较次数 :19900 比较所用时间为(ms)　：606

直接插入排序:

移动次数　：10465 比较次数 :199 比较所用时间为(ms)　：128

简单选择排序:

移动次数　：579 比较次数 :20100 比较所用时间为(ms)　：197

快速排序:

移动次数　：870 比较次数 :504 比较所用时间为(ms)　：43

希尔排序:

移动次数　：2586 比较次数 :1899 比较所用时间为(ms)　：58

堆排序:

移动次数　：4041 比较次数 :3895 比较所用时间为(ms)　：72

生成的数组为 : 111 123 161 124 164 137 124 102 34 90 182 26 37 174 151 136 88 180 121 90 129 37 137 151 47 40 46 172 142 41 24 5 164 137 129 80 74 53 182 60 95 117 86 132 91 190 20 131 122 141 173 3 131 110 155 178 151 153 150 93 195 126 98 111 63 27 192 137 32 174 197 80 43 36 164 134 178 185 18 52 126 191 55 9 102 10 139 5 116 42 50 63 168 148 174 32 127 118 121 111 45 71 143 88 59 108 175 37 93 193 89 171 184 144 181 38 107 120 43 175 162 45 190 83 193 116 67 72 35 140 136 32 11 79 120 70 139 47 59 32 40 148 156 177 45 89 167 104 161 163 31 76 8 21 159 154 137 178 178 172 118 114 156 130 146 77 152 85 124 12 70 117 112 26 46 109 67 13 165 28 128 196 56 137 17 167 43 107 145 21 31 16 136 188 98 82 17 50 119 93

冒泡排序:

移动次数　：31554 比较次数 :19900 比较所用时间为(ms)　：592

直接插入排序:

移动次数　：10900 比较次数 :199 比较所用时间为(ms)　：136

简单选择排序:

移动次数　：591 比较次数 :20100 比较所用时间为(ms)　：197

快速排序:

移动次数　：861 比较次数 :501 比较所用时间为(ms)　：43

希尔排序:

移动次数　：2446 比较次数 :1899 比较所用时间为(ms)　：56

堆排序:

移动次数　：4056 比较次数 :3910 比较所用时间为(ms)　：68

生成的数组为 : 14 189 10 127 167 8 188 34 174 154 15 102 102 71 191 72 191 34 179 136 56 10 152 144 150 50 178 167 53 97 61 67 39 23 146 6 32 135 193 6 41 8 60 143 31 52 15 22 38 146 111 94 109 15 190 59 66 168 27 119 66 40 138 57 63 85 63 47 172 8 5 13 16 66 108 0 70 124 174 108 22 85 155 131 101 145 191 119 66 170 190 84 10 80 141 73 165 156 73 89 165 78 54 181 144 163 133 166 39 60 27 61 145 182 193 198 79 136 69 145 106 11 181 116 92 122 141 9 79 166 99 44 45 105 177 141 68 63 108 59 123 135 73 20 69 18 171 100 154 40 198 12 4 179 80 96 102 21 105 133 188 156 129 185 62 106 126 82 169 186 142 44 73 167 65 142 185 36 195 139 28 193 151 32 172 31 80 26 4 138 111 144 46 40 129 60 147 8 143 68

冒泡排序:

移动次数　：27939 比较次数 :19900 比较所用时间为(ms)　：544

直接插入排序:

移动次数　：9697 比较次数 :199 比较所用时间为(ms)　：121

简单选择排序:

移动次数　：579 比较次数 :20100 比较所用时间为(ms)　：196

快速排序:

移动次数　：827 比较次数 :481 比较所用时间为(ms)　：40

希尔排序:

移动次数　：2563 比较次数 :1899 比较所用时间为(ms)　：62

堆排序:

移动次数　：4080 比较次数 :3916 比较所用时间为(ms)　：71

生成的数组为 : 194 37 113 68 4 130 10 189 118 157 80 98 150 31 83 75 14 163 53 170 101 165 115 100 5 196 160 104 4 103 173 199 92 38 67 48 120 29 189 38 187 69 136 89 52 19 164 18 135 18 189 188 183 104 88 140 100 49 45 57 104 170 56 197 160 123 197 80 104 187 118 91 8 6 181 61 178 97 79 113 115 68 101 50 124 190 191 177 191 188 34 95 158 90 44 118 165 42 198 69 181 68 113 189 26 46 50 4 143 82 69 11 102 171 61 27 113 4 156 104 192 190 199 102 32 196 20 149 190 170 170 171 38 83 160 17 129 163 21 25 197 91 36 99 14 49 78 79 54 34 183 198 176 134 101 8 82 121 109 72 44 80 43 34 163 156 51 45 119 25 22 116 68 58 167 82 59 46 113 113 80 96 64 9 182 165 169 65 38 79 137 82 159 133 117 74 89 120 71 160

冒泡排序:

移动次数　：30594 比较次数 :19900 比较所用时间为(ms)　：622

直接插入排序:

移动次数　：10590 比较次数 :199 比较所用时间为(ms)　：158

简单选择排序:

移动次数　：594 比较次数 :20100 比较所用时间为(ms)　：296

快速排序:

移动次数　：840 比较次数 :489 比较所用时间为(ms)　：41

希尔排序:

移动次数　：2668 比较次数 :1899 比较所用时间为(ms)　：60

堆排序:

移动次数　：4071 比较次数 :3923 比较所用时间为(ms)　：71

生成的数组为 : 145 93 76 13 103 43 47 163 41 160 28 74 8 92 83 143 9 4 160 48 83 97 82 194 182 199 69 71 72 92 31 17 186 59 183 89 55 30 4 96 143 33 170 103 125 5 46 135 10 6 135 45 56 17 40 38 169 61 62 41 153 93 58 91 105 41 133 160 24 137 56 119 170 179 22 48 136 21 135 146 179 70 144 187 87 184 178 56 197 40 97 102 85 108 146 190 101 79 150 77 16 159 196 187 90 171 187 26 144 122 125 123 192 69 63 31 5 41 88 154 81 137 56 166 197 2 109 51 33 59 128 50 170 77 189 60 0 176 39 144 50 164 19 42 33 82 25 190 123 113 144 156 3 152 123 152 107 184 3 140 43 132 142 166 161 131 26 161 59 17 105 61 181 76 103 166 159 81 108 34 146 52 191 149 157 66 102 64 50 57 156 45 141 99 11 102 182 190 63 194

冒泡排序:

移动次数　：26937 比较次数 :19900 比较所用时间为(ms)　：520

直接插入排序:

移动次数　：9371 比较次数 :199 比较所用时间为(ms)　：113

简单选择排序:

移动次数　：582 比较次数 :20100 比较所用时间为(ms)　：212

快速排序:

移动次数　：868 比较次数 :503 比较所用时间为(ms)　：44

希尔排序:

移动次数　：2515 比较次数 :1899 比较所用时间为(ms)　：78

堆排序:

移动次数　：4065 比较次数 :3923 比较所用时间为(ms)　：91

生成的数组为 : 7 120 55 141 197 111 107 156 192 16 190 138 20 133 40 177 199 142 41 1 151 198 47 93 49 10 195 31 0 59 25 8 179 81 149 128 144 8 84 136 176 27 74 197 112 114 174 112 8 168 65 160 118 112 5 167 123 0 150 75 11 128 35 143 161 136 71 105 145 108 41 121 135 67 70 47 182 197 111 142 165 177 54 35 41 59 154 116 60 104 192 23 32 27 166 145 116 38 50 61 146 43 134 33 111 157 32 45 154 96 139 119 73 194 154 66 5 108 183 17 164 127 41 149 106 159 94 22 197 97 83 95 140 170 80 3 127 113 0 81 9 140 152 34 86 58 52 91 118 187 61 82 114 102 183 21 61 78 43 11 175 79 58 67 1 139 23 128 4 23 161 165 115 65 151 153 123 3 197 41 143 58 75 57 112 59 78 125 137 74 88 64 153 147 83 154 38 58 34 42

冒泡排序:

移动次数　：31449 比较次数 :19900 比较所用时间为(ms)　：583

直接插入排序:

移动次数　：10873 比较次数 :199 比较所用时间为(ms)　：132

简单选择排序:

移动次数　：573 比较次数 :20100 比较所用时间为(ms)　：191

快速排序:

移动次数　：854 比较次数 :496 比较所用时间为(ms)　：40

希尔排序:

移动次数　：2595 比较次数 :1899 比较所用时间为(ms)　：59

堆排序:

移动次数　：4086 比较次数 :3930 比较所用时间为(ms)　：70

## 附录

主要程序:内部排序比较.cpp

次要的:名叫out.txt的文件