

总体评价



姜颖琪 按时通关

学号：U201915145

分班：计科1907

截止前完成关卡：4/4

最新完成关卡：4/4

完成效率：--

课堂最高完成效率：--

通关时间	计时规则?	实训总耗时?	评测次数	查重扣分	迟交扣分	最终成绩	总评
2022-07-08 16:52	页面停留时长	21天 7小时 27分钟 28秒	32	--	--	100.0/100.0	优秀

阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	串行环境下的排序算法	2022-06-14 13:55	25	10	2022-07-05 15:44	21天 1小时 48分钟 55秒	否	200/200	25.00/25	25.00
2	pthread环境下的并行排序算法设计	2022-06-14 13:55	108	20	2022-07-06 17:12	1小时 42分钟 4秒	否	300/300	25.00/25	25.00
3	OpenMP环境下的并行排序算法设计	2022-06-14 13:55	93	1	2022-07-06 17:23	11分钟 25秒	否	300/300	25.00/25	25.00
4	MPI环境下的并行排序算法设计	2022-06-14 13:55	235	1	2022-07-08 16:52	3小时 45分钟 4秒	否	300/300	25.00/25	25.00

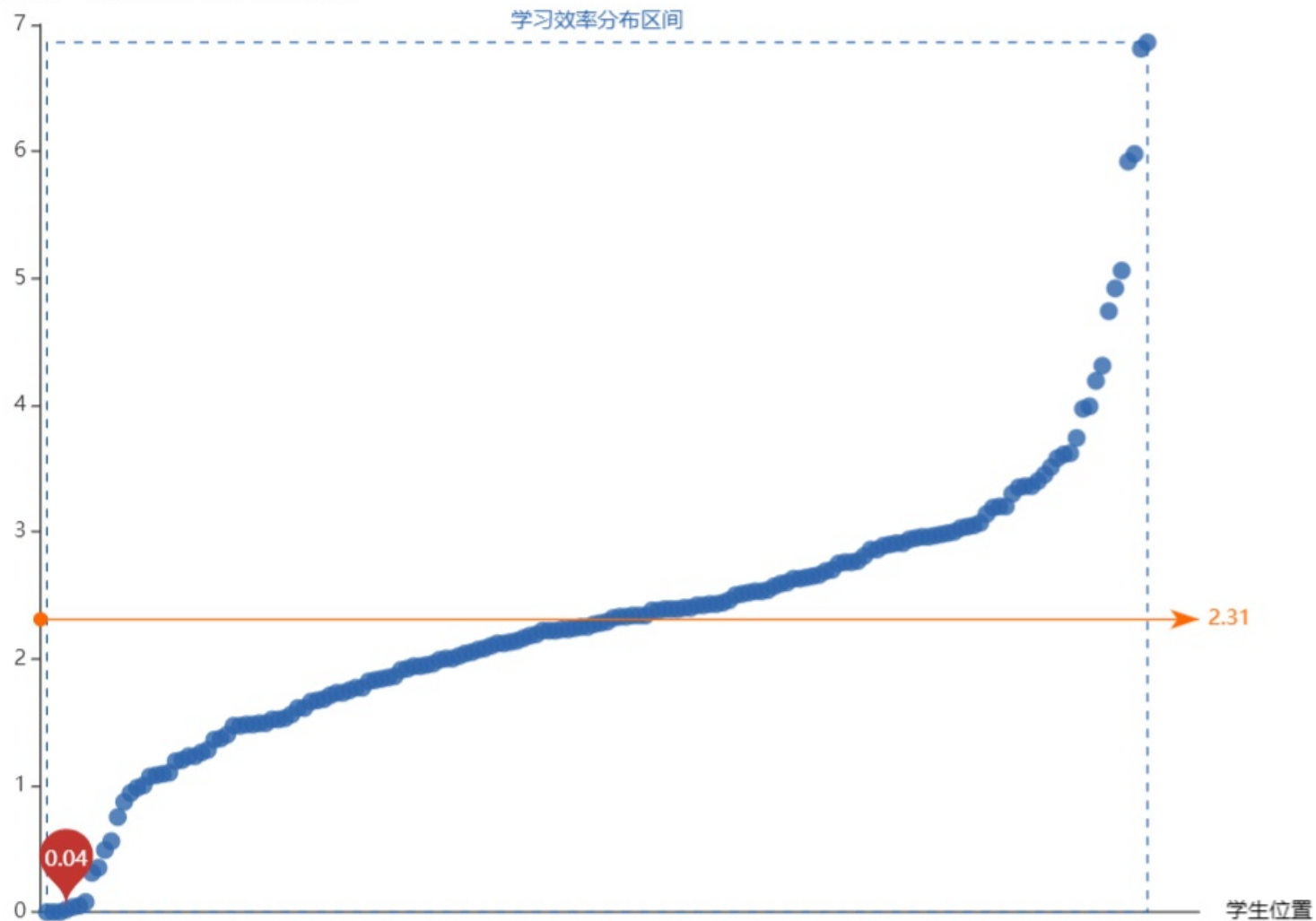
图形统计

学习效率

学习效率：-- 实训总得分：100.00 实训总耗时：21天 7小时 27分钟 28秒

姓名 姜颖琪

学习效率=log(实训总得分/实训总耗时)



学号	U201915145
学习效率	0.04
当前排名	168

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

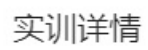
能力值

能力值 (实训获得经验值/实训评测次数)

姓名	姜颖琪
学号	U201915145
能力	34.38

1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少







最后通关代码 通过了本题，在作业时间内最后一次提交的通关代码

代码修改行数 25

src/ex1/ex1.c

```
1  #include <stdio.h>
2  struct ThreadArg
3  {
4      int* arr;
5      int start;
6      int end;
7  };
8
9  //思路:
10 //快排的原理为 先找到第一个数对应的位置，然后对两边分别排序，由于两边是单独的因此可以直接并行操作
11 /***** Begin *****/
12
13 int partition(int arr[], int low, int high){
14
15     int key;
16     key = arr[low];
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.24	34.52	2022-07-06 17:10	查看代码
2	共有4组测试集，其中有4组测试结果不匹配	0.24	34.52	2022-07-06 17:09	查看代码
3	共有4组测试集，其中有4组测试结果不匹配	0.24	34.52	2022-07-06 17:09	查看代码
4	评测通过	0.24	34.52	2022-07-06 10:43	查看代码
5	共有4组测试集，其中有4组测试结果不匹配	0.24	34.52	2022-07-06 10:43	查看代码
6	共有4组测试集，其中有4组测试结果不匹配	0.24	34.52	2022-07-06 10:42	查看代码
7	评测通过	0.24	34.52	2022-07-05 15:44	查看代码

8	共有4组测试集，其中有4组测试结果不匹配	0.24	34.52	2022-07-05 15:43	查看代码
9	共有4组测试集，其中有4组测试结果不匹配	0.24	34.52	2022-07-05 15:42	查看代码
10	共有4组测试集，其中有4组测试结果不匹配	0.24	34.52	2022-07-05 15:41	查看代码



最后通关代码 通过了本题，在作业时间内最后一次提交的通关代码

代码修改行数 108

src/ex2/ex2.c

```
1  #include <stdio.h>
2  #include <pthread.h>
3  #include <math.h>
4  #include <stdlib.h>
5  struct ThreadArg
6  {
7      int* arr;
8      int start;
9      int end;
10 };
11
12 //思路:
13 //快排的原理为 先找到第一个数对应的位置，然后对两边分别排序，由于两边是单独的因此可以直接并行操作
14 /***** Begin *****/
15
16 int partition(int arr[], int low, int high){
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.24	34.52	2022-07-06 17:12	查看代码
2	共有4组测试集，其中有4组测试结果不匹配	0.24	34.52	2022-07-06 17:12	查看代码
3	共有4组测试集，其中有4组测试结果不匹配	0.24	34.52	2022-07-06 17:08	查看代码
4	共有4组测试集，其中有4组测试结果不匹配	0.24	34.52	2022-07-06 17:06	查看代码
5	共有4组测试集，其中有4组测试结果不匹配	0.24	34.52	2022-07-06 17:06	查看代码
6	共有4组测试集，其中有4组测试结果不匹配	0.24	34.52	2022-07-06 17:03	查看代码
7	共有4组测试集，其中有4组测试结果不匹配	0.24	34.52	2022-07-06 16:28	查看代码

8	共有4组测试集，其中有4组测试结果不匹配	0.24	34.52	2022-07-06 16:13	查看代码
9	共有4组测试集，其中有4组测试结果不匹配	0.24	34.52	2022-07-06 15:55	查看代码
10	共有4组测试集，其中有4组测试结果不匹配	0.24	34.52	2022-07-06 15:54	查看代码



最后通关代码 通过了本题，在作业时间内最后一次提交的通关代码

代码修改行数 93

src/ex3/ex3.c



```
1  #include <stdlib.h>
2  #include <stdio.h>
3  #include <math.h>
4  #include <string.h>
5  struct ThreadArg
6  {
7      int* arr;
8      int start;
9      int end;
10 };
11
12 //思路:
13 //快排的原理为 先找到第一个数对应的位置，然后对两边分别排序，由于两边是单独的因此可以直接并行操作
14 /***** Begin *****/
15
16 int partition(int arr[], int low, int high){
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.24	34.52	2022-07-06 17:23	查看代码



最后通关代码 通过了本题，在作业时间内最后一次提交的通关代码

代码修改行数 235

src/ex4/ex4.c



```
1  #include <stdlib.h>
2  #include <stdio.h>
3  #include <mpi.h>
4  #include <math.h>
5  #include <string.h>
6  struct ThreadArg
7  {
8      int* arr;
9      int start;
10     int end;
11 };
12
13 // 顺序循环队列
14 // #define MaxSize 3    //定义队列中元素的最大个数
15 // typedef int ElemType;
16 // typedef struct{
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.24	34.52	2022-07-08 16:52	查看代码