End Semester July-August 2020

Subject: Algorithm

Subject Code: CS206

Duration: 01.08.2020 to 11.08.2020.

Submission deadline: 10.45 p.m., 12.08.2020

I. Prepare the report under the following heads (given in bold letters alongside each point) keeping in mind that the report should contain a detailed analysis of the solution:

a. Describe in your own words your overall project, i.e., what your mini project is about. (**Project Description**)

(1)

b. Give the detailed step by step algorithm to solve the problem given in your project. (The algorithm should be in the form of pseudocode obtained from the code you have written to solve the problem, hiding the implementation details that does not feature in the complexity analysis. Direct code will not be accepted as the algorithm.) There should be a detailed discussion of your algorithm. (**Algorithm**)

(5+3)

c. The input to the algorithm and desired output of the algorithm must be clearly specified at the beginning of the algorithm.

(1+1)

d. Prove that the algorithm is correct using loop invariant, mathematical induction, or any other logical argument. All statements and claims made must be proved. (**Proof of Correctness of the Algorithm)**

(5)

e. Justify that your solution to the given problem is correct. (If you had to make a choice among different algorithms, then you will have to discuss the drawbacks of the other sorting algorithms for solving the given problem. Or, if your solution is a modified version of the standard algorithm then you will have to compare the advantage of your modified version over the standard version with respect to the given problem.) (**Justification of Solution)**

(5)

f. Find the time complexity of the algorithm, given in (b), giving detailed cost of each step. Discuss the best-case and worst-case time complexities. If it's

a recursive algorithm, then give the recurrence relation and solve it using any method of solving recurrences of your choice. (**Time Complexity Analysis)** (4)

g. Take a suitable input instance of your choice and illustrate the stepwise running of your algorithm to obtain the final output. Show the advantage of your solution by comparing it with the other possible solutions if you had to make a choice of algorithm from a set of similar algorithms. (**Results and Discussion**) (3)

h. Discuss the drawback of your solution if any. (Possibly give an instance/type of input instance for which your solution will give incorrect output). (**Drawbacks**) (2)

II. Prepare the mini project assigned to you, as given in the table below, according to the guideline given above. Just giving the solution is not enough, you will have to justify it with proper arguments.

Mini Projects 80 to 89.

80. You want to place cell phone stations at certain points along a road, with houses scattered very sparsely along it, so that every house is within 10 kilometers of one of the base stations. Give an efficient algorithm that achieves this goal using as few base stations as possible.

81. CPU scheduling deals with the problem of deciding which of the processes in the ready queue (main memory) is to be allocated the CPU.
A different approach to CPU scheduling is the Priority Scheduling algorithm. A priority is associated with each process, and the CPU is allocated to the process with the highest priority. Equal-priority processes are scheduled in first-come, first-served (FCFS) order. Priorities are generally some fixed range of numbers, such as 0 to 7, or 0 to 4,095. However, there is no general agreement on whether 0 is the highest or lowest priority. Consider 0 to be the highest priority.
Write an algorithm to implement this CPU scheduling for a given set of processes $P = \{p_1, p_2, \dots, p_n\}$ having arrival time $A = \{a_1, a_2, \dots, a_n\}$ ($a_1 = a_2 = \cdots = a_n = 0$, i.e., all processes are there in the system at the same time), CPU burst time (running time of the process in the CPU) $B = \{b_1, b_2, \dots, b_n\}$, and the priorities $S = \{s_1, s_2, \dots, s_n\}$ respectively. The output of your algorithm should be the average waiting time for the processes.

(Waiting Time: The CPU scheduling algorithm does not affect the amount of timing during which a process executes or does I/O; it affects only the amount of time that a process spends waiting in the ready queue. Waiting time is the sum of the periods spent waiting in the ready queue.)

82. When multiple processes run in the memory, each process is allocated a fixed amount of memory space in the form of a stipulated number of frames. Let's consider a process that has been allocated 3 such memory frames (or pages). Thus, although the process requires many more pages or memory frames, only three of its active pages can reside in the main memory at any time. The process in its course of execution makes the following sequence of page references (reference string):
    1, 2, 1, 3, 7, 4, 5, 4, 5, 6, 3, 1
    If now a request is made for a page of the process which is not residing in the main memory then a page fault is said to occur and a victim page has to be selected and replaced by this newly demanded page. The page that is replaced depends on the page replacement policy used. The performance of the page replacement algorithm is measured by the number of page faults occurring as they require secondary memory access.
    Hence, write an algorithm to find the number of page faults occurring when FIFO page replacement policy is used, which is **replacement of the oldest page**. How many page faults occur for the above reference string? Give step-wise output for the given input by showing the status of the 3 page-frames at each step. Also, find the complexity analysis of your algorithm considering a reference string of size n and process has been allocated m frames. Give details of the data structure used.
83. When multiple processes run in the memory, each process is allocated a fixed amount of memory space in the form of a stipulated number of frames. Let's consider a process that has been allocated 3 such memory frames (or pages). Thus, although the process requires many more pages or memory frames, only three of its active pages can reside in the main memory at any time. The process in its course of execution makes the following sequence of page references (reference string):
    1, 2, 1, 3, 7, 4, 5, 4, 5, 6, 3, 1
    If now a request is made for a page of the process which is not residing in the main memory then a page fault is said to occur and a victim page has to be selected and replaced by this newly demanded page. The page that is replaced depends on the page replacement policy used. The performance of the page replacement algorithm is measured by the number of page faults occurring as they require secondary memory access.

Hence, write an algorithm to find the number of page faults occurring when LRU page replacement policy is used, which is **replacement of the page that has not been used for the longest period of time**. How many page faults occur for the above reference string? Give step-wise output for the given input by showing the status of the 3 page-frames at each step. Also, find the complexity analysis of your algorithm considering a reference string of size n and process has been allocated m frames. Give details of the data structure used.

84. In any base $b \geq 2$, the sum of any three single-digit numbers is at most two digits long. Write an algorithm with the help of which you can empirically prove the above statement. Also give a theoretical proof for it.

85. CPU scheduling deals with the problem of deciding which of the processes in the ready queue (main memory) is to be allocated the CPU.
The Round-Robin (RR) scheduling algorithm is designed especially for time sharing system. A small unit of time called a time quantum (or time slice), is defined. A time quantum is generally from 10 to 100 milliseconds. The CPU scheduler goes around the ready queue, allocating the CPU to each process for a time interval of up to one quantum of time till the process ends.
Write an algorithm to implement the Round-Robin (RR) scheduling algorithm Preemptive Priority Scheduling algorithm for a given set of processes $P = \{p_1, p_2, \dots, p_n\}$ having arrival time $A = \{a_1, a_2, \dots, a_n\}$, and the CPU burst time (running time of the process in the CPU) $B = \{b_1, b_2, \dots, b_n\}$ respectively. The output of your algorithm should be the average waiting time for the processes.
(Waiting Time: The CPU scheduling algorithm does not affect the amount of timing during which a process executes or does I/O; it affects only the amount of time that a process spends waiting in the ready queue. Waiting time is the sum of the periods spent waiting in the ready queue.)

86. Given a sequence $a_1, a_2, \dots, a_n$ and $b_1, b_2, \dots, b_n$ where $a_i > 0, b_i > 0$ are the processing power of machines A and B in the ith minute respectively. Given a sequence of n minutes, a plan is specified by a choice of A, B, or 'move' for each minute, with the property that choices A and B cannot appear in consecutive minutes. For example, if your job is on machine A in minute i, and you want to switch to machine B, then your choice for minute $i + 1$ must be 'move' and then your choice for minute $i + 2$ can be B. The value of a plan is the total number of steps that you manage to execute over the n minutes: so it's the sum of a_i over all minutes in which the job is on A plus the sum of b_i over all minutes in which the job is on B. Give an efficient

algorithm that takes values for $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$ and returns the value of an optimal plan (i.e., a plan of maximum value).

87. When multiple processes run in the memory, each process is allocated a fixed amount of memory space in the form of a stipulated number of frames. Let's consider a process that has been allocated 3 such memory frames (or pages). Thus, although the process requires many more pages or memory frames, only three of its active pages can reside in the main memory at any time. The process in its course of execution makes the following sequence of page references (reference string):
    1, 2, 1, 3, 7, 4, 5, 4, 5, 6, 3, 1
    If now a request is made for a page of the process which is not residing in the main memory then a page fault is said to occur and a victim page has to be selected and replaced by this newly demanded page. The page that is replaced depends on the page replacement policy used. The performance of the page replacement algorithm is measured by the number of page faults occurring as they require secondary memory access.
    Hence, write an algorithm to find the number of page faults occurring when MRU page replacement policy is used, which is **replacement of the most recently used page**. How many page faults occur for the above reference string? Give step-wise output for the given input by showing the status of the 3 page-frames at each step. Also, find the complexity analysis of your algorithm considering a reference string of size n and process has been allocated m frames. Give details of the data structure used.

88. When multiple processes run in the memory, each process is allocated a fixed amount of memory space in the form of a stipulated number of frames. Let's consider a process that has been allocated 3 such memory frames (or pages). Thus, although the process requires many more pages or memory frames, only three of its active pages can reside in the main memory at any time. The process in its course of execution makes the following sequence of page references (reference string):
    1, 2, 1, 3, 7, 4, 5, 4, 5, 6, 3, 1
    If now a request is made for a page of the process which is not residing in the main memory then a page fault is said to occur and a victim page has to be selected and replaced by this newly demanded page. The page that is replaced depends on the page replacement policy used. The performance of the page replacement algorithm is measured by the number of page faults occurring as they require secondary memory access.
    Hence, write an algorithm to find the number of page faults occurring when optimal page replacement policy is used, which is **replacement of the page**

that will not be used for the longest period of time. How many page faults occur for the above reference string? Give step-wise output for the given input by showing the status of the 3 page-frames at each step. Also, find the complexity analysis of your algorithm considering a reference string of size n and process has been allocated m frames.

89. A server has n customers waiting to be served. The service time required by each customer is known in advance: it is t_i minutes for customer i. So if, for example, the customers are served in order of increasing i, then the ith customer has to wait $\sum_{j=1}^{i} t_j$ minutes.

We wish to minimize the total waiting time

T = $\sum_{i=1}^{n}$(time spent waiting by customer i).

Give an efficient algorithm for computing the optimal order in which to process the customers.

Roll No.	Assigned Mini Project Number
1815002	89
1815057	80
1815059	86
1815078	87
1815113	88
1815044	84
1815086	85
1815095	83
1815110	81
1815118	82